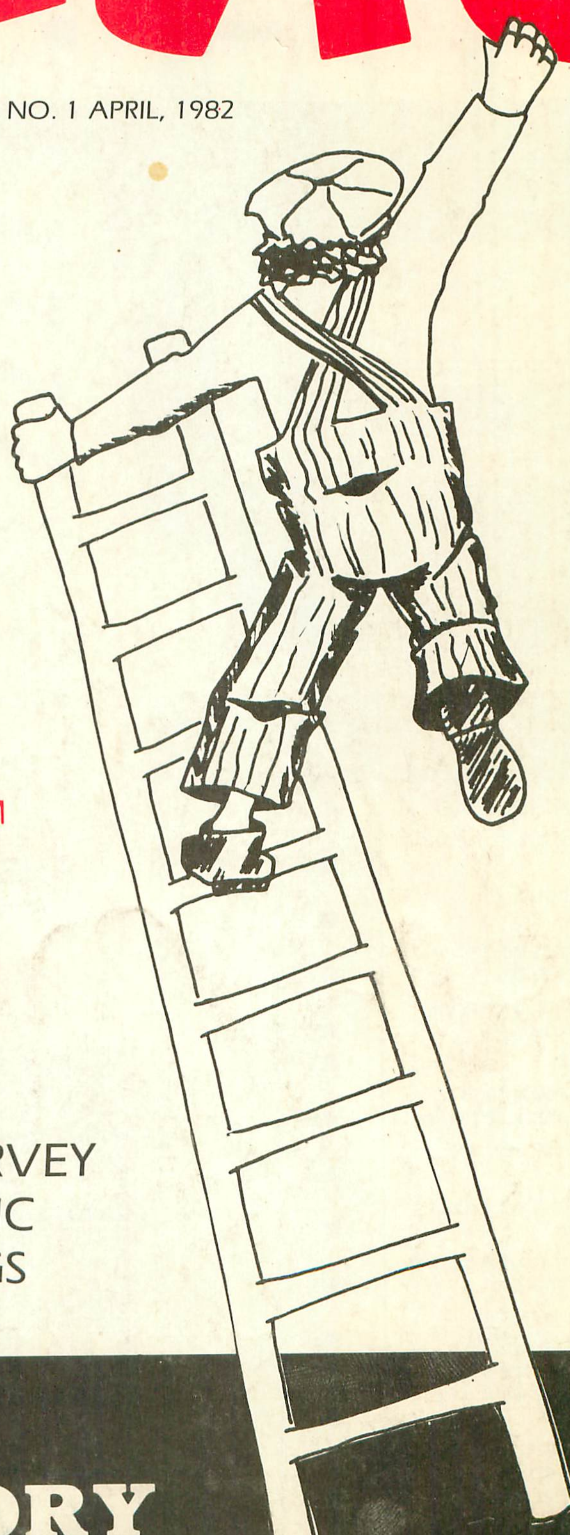
Antic™

VOL. 1, NO. 1 APRIL, 1982



The ATARI™ Resource

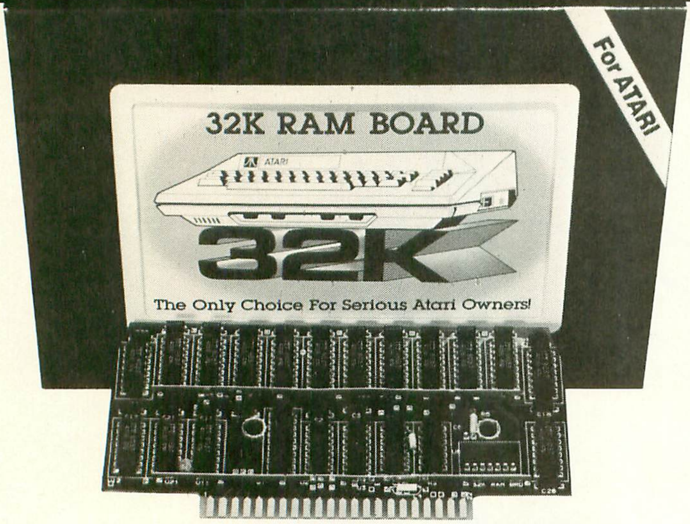
INSIDE:
WORD PROCESSOR SURVEY
PILOT — FORTH — BASIC
FIVE PROGRAM LISTINGS
MEMORY MAP

INTRODUCTORY ISSUE

$2.50

THE MOSAIC 32K RAM FOR ATARI

THE BEST:



✓ DESIGN
✓ MATERIALS
✓ SAFETY
✓ GUARANTEE

THE ONLY CHOICE

You own the best micro computer available, the Atari*. At Mosaic we've made Atari computers our only business and have made our products the best anywhere. You've seen the advantage of having a 32K RAM board. The Mosaic 32K RAM is the only board designed to meet your needs now and in the future too. It has designed-in flexibility to be compatible with products available in the near future. See for yourself, Mosaic is the only choice for the serious Atari owner.

THE BEST SCREEN CLARITY

Here's what A.N.A.L.O.G. magazine had to say: "Mosaic uses, what we feel are the highest quality components, which may improve reliability." and "The Mosaic showed no sign of interference and gave the best screen clarity."

THE MOSAIC ADVANTAGE	MOSAIC 32K RAM	OTHER 32K RAMS
Works in both Atari 400 & 800	■	■
Gold edge connectors for better reliability	■	■
Fits Atari 400 without modification	■	■
Custom components for better performance & reliability	■	
Highest quality components for the best screen clarity	■	
Four year warranty	■	
Designed to take advantage of Atari 800's superior bus structure.	■	
Can be used with 8K, 16K and future products.	■	
Allows Atari 800 to have 2 slots for future expansion	■	
Designed so there's no danger of damaging your computer	■	
Designed for inter-board communication in Atari 800	■	
Easy to follow instructions for simpler no-solder installation in Atari 400	■	
Available companion board ($5) to allow running 32K board independent of other boards	■	
Full flexible memory configuration	■	

Atari 800 Memory Configuration	with the MOSAIC 32K RAM	with other 32K Boards
Empty / 32K / 16K	48K RAM 40K With BASIC Cartridge	48K RAM 40K With BASIC Cartridge
Empty / 32K / 8K	40K RAM	**Danger!** This Configuration Can Damage Computer
Empty / 16K / 32K	48K RAM 40K With BASIC Cartridge	**Danger!** This Configuration Can Damage Computer
Empty / 8K / 32K	40K RAM	**Danger!** This Configuration Can Damage Computer
8K / 32K / 8K	48K RAM 40K With BASIC Cartridge	**Danger!** This Configuration Can Damage Computer
Empty / Companion / 32K	32K RAM	**Danger!** This Configuration Can Damage Computer

Now from your nearest Mosaic dealer
800 547-2807

$179.95

Direct from Mosaic $189.95

MOSAIC ELECTRONICS
P.O. Box 748 Oregon City, Oregon 97045 503/655-9574

*Trade Mark of Atari, Inc.

Antic™

The ATARI™ Resource

ANTIC
Publishing

Editor & Publisher
James Capparell

Associate Editor
Gary Phillips

Assistant Editor
DeWitt Robbeloth

Products Review Editor
Jon Loveless

Production and Art
Marni Tapscott

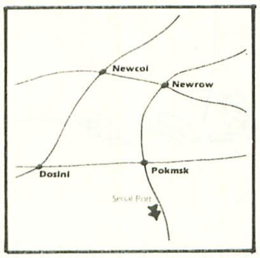
Business Manager
Karen Phillips

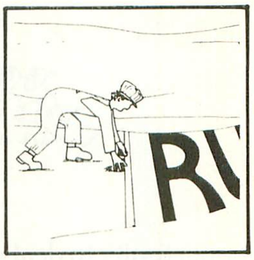
Contributing Writers
Ken Harms
Bob Gonsalves
Bill Wilkenson
Matt Loveless

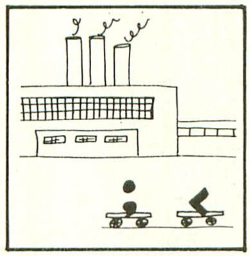
ANTIC *encourages writers and programmers to submit work for publication. Writers warrant that their work is original and ownership resides with the author. Material not accepted for publication will be returned provided that sufficient postage is enclosed. Authors acknowledge that material accepted for publication becomes the exclusive property of* **ANTIC** *Publishing.*

April, 1982, Volume 1, Number 1 **ANTIC** *- The ATARI Resource is published six times per year by* **NTIC** *Publishing. Editorial offices located at 297 Missouri Street, San Francisco, Ca. 94107. Phone is (415) 864-0886. Domestic subscriptions are $15 for six issues; $27 for 12 issues. Mail subscription orders with check or money order to above address.*

TABLE OF CONTENTS







EDITORIAL



This page will give me an opportunity to deal with pressing topics, and a chance to speak directly to you our readers. I won't be writing this department every month—just whenever something useful presents itself. For this issue the most pressing topic I can think of is who we are and what we plan for the coming year.

We are dedicated Atari enthusiasts who have been involved with the 800 and 400 since it came on the market back in late 1979. We have varied backgrounds, some of us are computer professionals and some of us are what I've come to call Atari professionals, very motivated Atari users. What we have in common is our enthusiasm for home computing, especially using Atari, and our desire to share this enthusiasm.

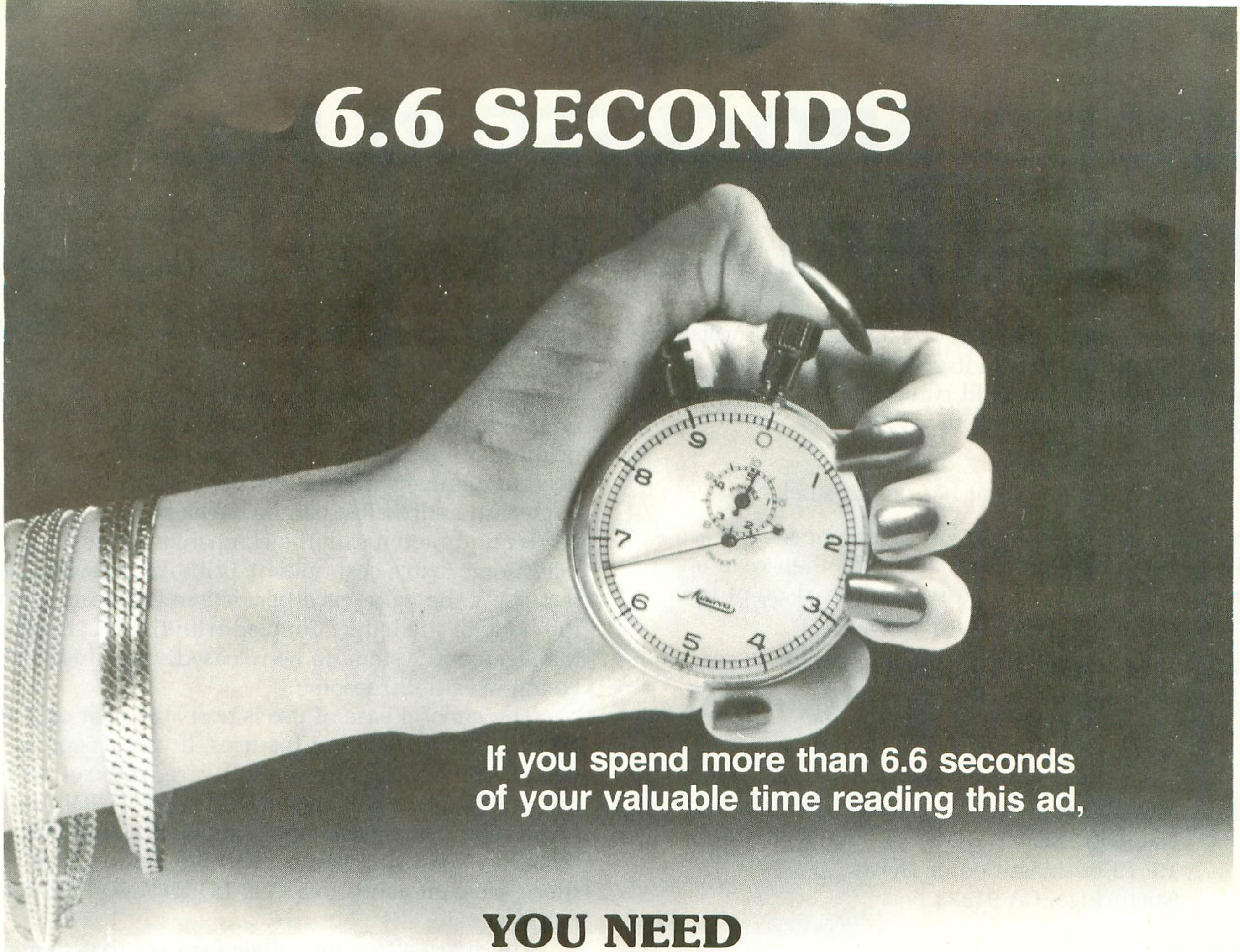
We've looked carefully at the Atari user community and watched it grow until now there are approximately 100,000 users. Most of us at the magazine have been involved in computer clubs and newsletter publications. We now think it's time Atari users had their own dedicated publication, a resource which can be turned to for help and up-to-date information. We intend to create a publication that you, as Atari users, can be proud of; a journal that will keep you abreast of the latest hardware and software developments; a national forum with articles written by acknowledged experts and motivated beginners; a magazine you will find indispensable. We intend to bring you the best reviews, the most informative how-to-do-it articles, and perhaps an insiders' column from Atari headquarters. Additionally, we shall make available the collected volumes of public domain software and this at a price you can't possibly ignore. (See our application for further information.) We have scheduled six issues in this our first year. Expect to see some changes in format and content as we learn who you are and what you need. Actually, that's my clever way of asking you for your support by way of phone calls and letters and of course your subscriptions.

We hope to provide a publication you shall all look forward to receiving. For beginners we have Starting Line, advanced programmers see Systems Guide, and software publishers and would-be publishers watch for our interviews with some Atari programmers who are making it. There's more, so keep watching—with your support we're really going to GROW.

Oh, one more thing. ANTIC will finally give Atari enthusiasts a strong clear voice that will be heard down there in Silicon Valley.

Well, that's it. If you have any questions just drop us a note—we'll respond in our I/O Board (that's our input-output department).

James Capparell
Editor/Publisher
ANTIC
The ATARI Resource



YOU NEED

SpeedRead+

- Optimized Systems Software, the company that brought you BASIC A+ and OS/A+, proudly presents another *"PLUS"*.
- ***SpeedRead+*** is the *world's first* speed reading tutor designed for use on your personal computer.
- ***SpeedRead+*** *begins* with training your eyes and mind to function as the incredible precision machine they were meant to be.
- ***SpeedRead+*** goes beyond mere words — it trains you to recognize phrases and columns instantly — it exercises your peripheral vision — it increases your comprehension.
- ***SpeedRead+*** matches *your* pace — now and in the future.

AVAILABLE NOW!

For 16K ATARI® computers with disk, Introductory Price **$59.95**
Coming soon for APPLE II®. Inquire about availability of TRS-80® and cassette versions.

SEE YOUR DEALER TODAY!

Call or write for a descriptive brochure of ***SpeedRead+***™ and other fine OSS products, such as OS/A+, BASIC A+, and TINY C®.

ATARI, APPLE II, TRS-80, and TINY C are trademarks of Atari, Inc., Apple Computer, Inc., Radio Shack, and Tiny C Associates, respectively. ***SpeedRead+*** is a trademark of Eagle Software and Optimized Systems Software.

Optimized Systems Software, Inc., 10379 Lansdale Ave., Cupertino, CA 95014, (408) 446-3099

WORD PROCESSORS FOR THE ATARI A COMPARISON

by Jon Loveless

There are currently three true word processors available for the ATARI, and that in itself creates a few problems. Typically, you would run out and buy one the moment one of 2 things occur:

1. you have the money, or
2. you find a store that has one in stock.

If you have not yet done so, or have done so and are still interested in what else is available, then read on. We are going to do an exhaustive comparison of the currently available word processors.

NAME: *Letter Perfect* (TM) Vers. 2.0
by: LKJ Enterprises, INC.
P.O. Box 10827
St. Louis, MO 63129

NAME: *Text Wizard* (TM)
by: DataSoft INC
19519 Business Center Drive
Northridge, CA 91324

NAME: *Atari Word Processor*
by: Atari, Inc.
Borregas Drive
Sunnyvale, CA 94086

Note: *LKJ has 2 disk based versions of* Letter Perfect *and 1 ROM version. Version 1.0 was released early in 1981, and has been replaced by Version 2.0. Disk based Version 2.0 is the one being compared here and you should be aware of which version you are looking at before you buy.*

The most difficult part of any software comparison is the final recommendation, so we are going to present the recommendation first. The bottom line is that all three of these word processors are excellent. If you were to choose one at random with no advance knowledge about its capabilities, you would end up with an excellent product and a purchase of value. All three have a number of commands to get used to, and with practice can be mastered in fairly short order. After several sessions at the keyboard these commands will become second nature and a more detailed evaluation can be made. Here again, none will fail, but regardless of which you purchase you will undoubtedly find something you wish you had.

Having started with relatively equal booking, let's look at the best and worst features of each word processor first, and then fill in the middle ground with a chart of feature comparisons.

LETTER PERFECT (TM)

BEST FEATURES:

The database merge capability stands out as truly unique feature of *Letter Perfect*. This will allow the printing of form letters to a list of people without the need to manually change the name and address. The name and address list can be developed on *Letter Perfect* and then, using the companion program *Mail/Merge* (extra-cost option), print out individual letters using the name or other data in the heading and the body of the letter. Coupled with the LKJ Database Management System (to be released soon) this should be a powerful threesome.

The general ease of use is probably next on the list of *Letter Perfect* best features. If you accept the default values built into the software, you can boot the disk (a menu will appear), press [RETURN], and begin typing. It is really that simple. Pressing the [ESC] key will bring you back to the menu where there are commands to SAVE, LOAD AND PRINT, to name a few.

There are also a few other features that should come under the category of "nice touch". For example, pressing the CONTROL key and [A] will move the cursor to the beginning of the line you are on, and CONTROL [Z] will move the cursor to the end of the same line. This may sound insignificant, but with nearly 50 commands, and 15 format representations this kind of simple and logical interaction is appreciated.

WORST FEATURES:

Version 1.0 had a number of bugs, and other problems often associated with the first release of new software. Version 2.0 has corrected all of these and added a few other niceties. As a result, we are hard pressed to find what we could call a "worst." LKJ chose to utilize their own Disk Operating System which creates instant compatibility problems problems. This has been correctd to a large degree with the optional program *Mail/Merge*. In the absence of the comparison program, however, it does limit you to using *Letter Perfect* files. Atari's 40 column TV screen limitation is overcome somewhat by a "SCREEN FORMAT" command on the menu, but this is at best only a partial solution.

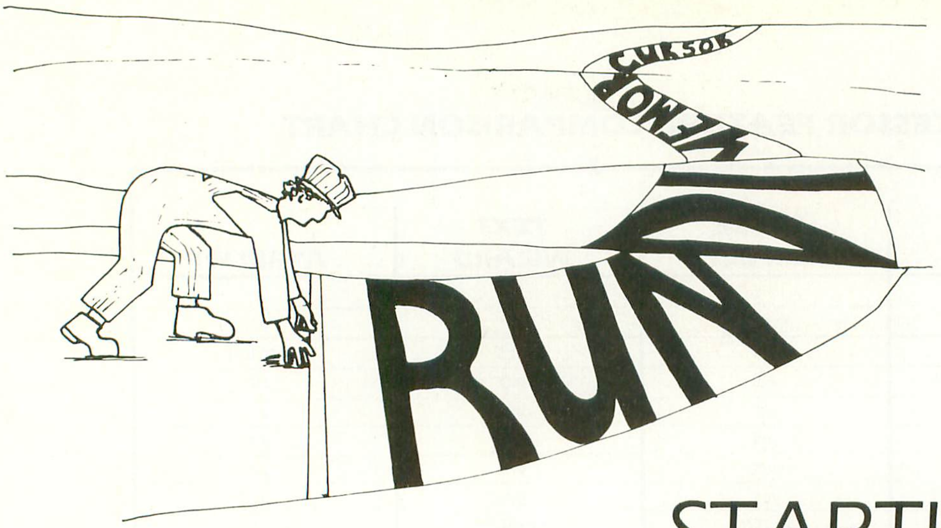
All in all, *Letter Perfect* is a well written, well engineered, easy to use program that will satisfy the vast majority of word processing needs.

continued on page 34

WORD PROCESSOR FEATURE COMPARISON CHART

FEATURE	LETTER PERFECT	TEXT WIZARD	ATARI WP
Printing			
Printer-825	yes	yes	yes
Printer-MX80	yes	yes	limited
multiple copy print	yes	no	no
single sheet print	yes	yes	yes
5 cpi	yes	yes	yes
10 cpi	yes	yes	yes
16 cpi	yes	yes	yes
proportional	yes-825	yes-825	yes-825
center text	yes	yes	yes
right margin text	yes	yes	yes
headers/page numbers	yes	yes	yes
footers/page numbers	yes	yes	yes
underline text	yes-825 (1)	yes-825 only	yes-825
pause print	no	no	no
double column print	no	yes-825 only	yes
Editing			
ATARI screen editor	yes	yes	yes
text parsing	yes	yes	yes
tab spacing	yes	yes	yes
adjustable tab	yes	no	yes
line spacing	yes-CR	yes-CR	yes-(2)
search text	yes	yes	yes
replace text	yes	yes	yes
selective srch/repl	no (3)	yes	yes
delete text	wide variety	wide variety	wide variety
move text	yes	yes	yes
go to end of text	yes	yes	yes
go to top of text	yes	yes	yes
on screen formatting	yes (4)	no	yes
text review (scroll)	yes	yes	yes
File Handling			
DOS used	LKJ (5)	ATARI	ATARI
disk formatting	yes	no	yes
lock/unlock	yes	no	yes
delete files	yes	yes	yes
save file-compacted	yes	yes	yes
merge files	yes	yes	yes
access database	yes (6)	no (7)	no
General			
memory limits	36714	30505	N/A (8)
memory left message	yes-update	yes-keystroke	yes-symbol
menu usage	one	none	5 plus
disk directory	yes	yes	yes

(1) also with MX80 if Graftrax equipped.
(2) Atari WP offers 3 methods of line insert.
(3) does not support search and replace/no-replace choice.
(4) screen formatting intersperses lines to simulate a page.
(5) with Mail/Merge ATARI DOS compatibility is available.
(6) LKJ database system to be released soon.
(7) compatibility with Filemanager 800 to be supported.
(8) Atari WP is page oriented—saved on a single page basis.



STARTING LINE

by James Capparell

If you are an ATARI owner new to the world of computing, you probably feel overwhelmed by the mass and complexity of the technical information necessary just to use your new computer. With your ATARI you received operating instructions for each piece of hardware. You have the BASIC Reference Manual and the BASIC self-teaching guide. At the store you saw a bewildering assortment of magazines, books and software! Where should you begin?

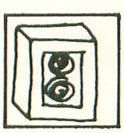
We here at **ANTIC** want you to know we realize your dilemma. Our magazine, and especially this column, should help you sort things out. **STARTING LINE** will be written in simple English. We are starting with some glossary items, and are committed to developing this department and glossary. Please let us know what you want to see here. If something is puzzling you, ask.

Our more advanced readers can help too. Send us those items that have been helpful in your own learning process. We'll all be grateful.

GLOSSARY



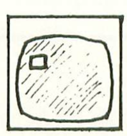
CARTRIDGE—This term is probably familiar to those of you with video recorders or tape recorders. In our case CARTRIDGE refers to that plastic box with BASIC printed on it. This small unit fits into the available slot on your 400 or 800. A CARTRIDGE contains a kind of computer MEMORY. This memory, unlike RAM mentioned earlier, has data(mail) stored in it only once at the factory. The computer can go read the data(mail) many times, but the contents of the mailbox will always be the same. These cartridges are referred to as ROM Read Only Memory. Information can only be read from MEMORY, data is delivered or stored just once at the factory. This special kind of memory doesn't need a power source to remember data, the data is permanently stored in the cartridge.



CASSETTE RECORDER—This is nothing but a slightly modified, garden variety tape recorder. In big computer systems this would be called a magnetic tape storage unit. It's used to keep a permanent recording of your favorite game program or data. If you want to use and reuse a program then you must save it on some permanent storage, such as a cassette tape.



CONSOLE—This is probably a familiar term. It's used to refer to that part of your ATARI with the typewriter like keys and the sockets for various cables. The CONSOLE is the heart of your computer system. Inside are the various integrated circuits that have made home computers possible. CONSOLE is synonomous with the computers logic circuitry, MEMORY, and keyboard.



CURSOR—This strange word means pointer or indicator. The CURSOR is the white rectangle that appears on your T.V. screen whenever you turn your computer CONSOLE on. It really has a very useful function, it shows you where the next character will appear when you press a key. This is useful to know, as a few minutes sitting at your CONSOLE will demonstrate.

continued on page 8

continued from page 6



MEMORY—Imagine all the mailboxes on your street standing in single file next to each other. Each mailbox has an individual address so that when I say deliver mail to #329 you should be able to find it. Computer MEMORY is similar to this row of mailboxes. Each MEMORY location has its own address and the ability to hold data. Additionallym you may put(store) mail in a box(memory location) or take(read) mail from a box(memory location). Most importantly, this reading or storing of mail(data) may be performed in any order—randomly. So, reading or storing data into numbers 919, 105, 3, 1000 is perfectly O.K. This is often referred to as Random Access Memory (RAM). Your computer has many of these mailboxes, as many as 65536 addressable locations. Each location is capable of receiving data and storing the data until needed or until power is turned off.



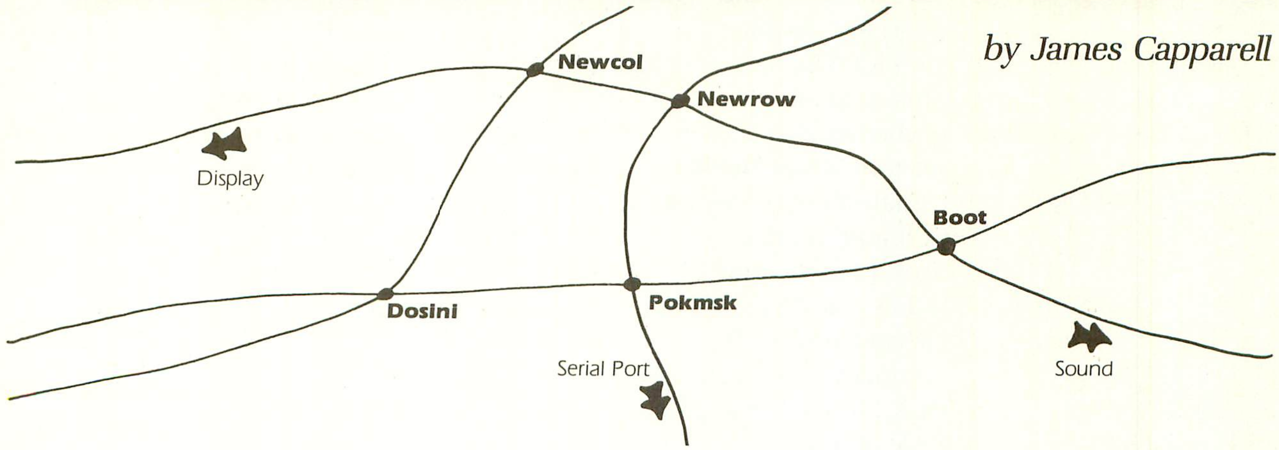
PROGRAM—also—**COMPUTER PROGRAM**—These terms indicate the special sequence of instructions that makes the circuitry inside the CONSOLE behave according to your desire. The instructions or program are delivered to MEMORY to be read later by the circuitry in the CONSOLE. The instructions tell the circuitry what to do. It's the CASSETTE RECORDER which will save the special sequence of instructions known as the PROGRAM from day to day.



CONTROL KEY—This key appearing next to the letter "A" on your keyboard is the abbreviation for CONTROL. When pressed it makes extra functions available. Usually you must press CTRL simultaneously with another key to enable the special function. For example, press CTRL and the key with the up arrow (next to the letter "P"). This will cause the CURSOR to move up one line.

SYSTEMS GUIDE
MEMORY MAP—PAGE 0

by James Capparell



This department will be devoted to systems related questions, descriptions, and elaborations. Over the next few issues of **ANTIC** we will construct a complete memory map. A memory map is nothing but a list of the reserved memory locations in RAM and ROM with a description of the purpose of each location.

We will present the addresses in ascending order with decimal and hexadecimal (base 16) address equivalent. Listed next to the address will be the number of contiguous locations that are related by function. Lobyte-hibyte address vectors are typical examples of two continguous bytes referred to by one name. Printed next to the number of locations will be the name assigned the address(es) in the ATARI documentation. Finally we give a simple (and sometimes not so simple) explanation of the function.

For example: 12 $C 2DOSINI
address 12 (decimal) or C (hexidecimal) has 2 continguous locations devoted to the function named DOSINI.

This month, in **SYSTEM GUIDE**, we describe the low end of RAM memory. These RAM locations are initialized whenever you turn the ATARI on. The values are copied from the ROM data base. Addresses 0 to 255 decimal are known as page 0. We are only showing the first 128 locations, those we've deciphered so far. I would like to invite all of you to participate in this project. Help us pry into the secrets of these last 128 bytes. If you've discovered something interesting, don't keep it a secret, share it. Our intent is to provide the most accurate up-to-date information. If you see an error, have an addendum, or can provide an interesting example please forward it and we'll all be doing one another a service. THANKS.

0 $0 1 LINZBS	???
1 $1 1 LINZBS	???
2 $2 2 CASINI	If cassette booted successfully during powerup then JSR thru here.
4 $4 2 RAMLO	Ram pointer for memory test.
6 $6 1 TRAMSZ	Temporary register for RAM size.
7 $7 1 TSTDAT	RAM test data register.
8 $8 1 WARMST	Warmstart flag set true (−1) when S/RESET pushed. When (=0) false then powerup retry.
9 $9 1 BOOT	Boot flag success indicator.
10 $A 2 DOSVEC	Disk software start vector.
12 $C 2 DOSINI	Used to store address of initialization of application upon DOS boot JSR indirect thru here to initialize application.
14 $E 2 APPMHI	Contains highest address of RAM needed by user. Screen handler opens S: only if no RAM needed below this address.
16 $10 1 POKMSK	IRQ service uses and alters POKMSK. These are POKEY interrupts. Shadow for IRQEN[$D20E]. bit 7=1 Break key interrupt enable. bit 6=1 Other key interrupt enable.

continued on next page

	bit 5=1 Serial input data ready interrupt enable. bit 4=1 Serial output data needed interrupt enable. bit 3=1 Serial out transmission finished interrupt enable.bit 2=1 Timer 4 interrupt enable. bit 1=1 Timer 1 interrupt enable.
17 $11 1 BRKKEY	This is initalized to -1 by OS.(1-=no break key pressed Monitored by keyboard also screen editor. Break during I/O returns status of $80. This is set to 0 when break key is pressed.
18 $12 3 RTCLOK	Updated every vblank interrupt (1/60 SEC.). Called frame counter initialized to 0 and overflows to 0. The least significant byte of counter is $12 and it uses 16msec units.
21 $15 2 BUFADR	Indirect buffer address register. Used as temporary page 0 pointer to current disk buffer.
23 $17 1 ICCOMT	Command for vector ????
24 $18 2 DSKFMS	Disk file manager pointer ????
26 $1A 2 DSKUTL	Disk utilities pointer ????
28 $1C 1 PTIMOT	Printer timeout every printer status request. Typical timeout for the 825 is 5 seconds. Initialized to 30 sec.
29 $1D 1 PBPNT	Print buffer pointer, index into printer buffer ranges from 0 to value of PBUFSZ.
30 $1E 1 PBUFSZ	Print buffer size of printer record for current mode. normal=40 bytes double width=20 bytes sideways=29 bytes status=4
31 $1F 1 PTEMP	Printer handler uses this temp register to save value of character to output to printer.
32 $20 1 ZIOCB	IOCBAS ICHIDZ HANDLER INDEX # FF=FREE IOCB
33 $21 1 ICDNOZ	Device # (DRIVE #)
34 $22 1 ICCOMZ	Command code
35 $23 1 ICSTAZ	Status of last IOCB action.
36 $24 1 ICBALZ	Buffer address low byte.
37 $25 1 ICBAHZ	Buffer address high byte.
38 $26 1 ICPTLZ	Put byte routine (address −1) low byte.
39 $27 1 ICPTHZ	Put byte high byte.
40 $28 1 ICBLIZ	Buffer length low byte.
41 $29 1 ICBLHZ	Buffer length high byte.
42 $2A 1 ICAZ1Z	Auxiliary information first byte.
43 $2B 1 ICAX2Z	Auxiliary information second byte.
44 $2C 1 ICSPRZ	Spare bytes local CIO use.
45 $2D 1 ICSPRZ	Spare bytes local CIO use.
46 $2D 1 ICSPRZ	IOCB Number multiplied by 16.
47 $3F 1 CIOCHR	Character byte for current operation.
48 $30 1 STATUS	Internal status storage.
49 $31 1 CHKSUM	Single byte sum with carry to least significant bit.
50 $32 1 BUFRLO	Pointer to data buffer low byte.??? which buffer.
51 $33 1 BUFRI	Pointer to data buffer high byte.
52 $34 1 BFENLO	Next byte past end of data buffer. (lobyte)
53 $35 1 BFENHI	Next byte past end of data buffer. (hibyte)
54 $36 1 CRETRY	Number of command frame retries.
55 $37 1 DRETRY	Number of device retries.
56 $38 1 BUFRFL	Buffer full flag.

57 $39 1 RECVDN	Receive done flag.
58 $3A 1 XMTDON	Transmission done flag.
59 $3B 1 CHKSNT	Checksum sent flag.
60 $3C 1 NOCKSM	No checksum follows data flag.
61 $3D 1 BPTR	Cassette record data index into data portion of record being read or written values range 0 to current value BLIMI [$28A] when BPTR=BLIM then buffer CASBOFF [$3FD] is empty if reading or full if writing
62 $3E 1 FTYPE	Interrecord Gap type. Copy of ICAX2Z from open command FTYPE 0 normal gaps. FTYPE 0 continuous gaps.
63 $3F 1 FEOF	Cassette end of file flag used by cassette handler to indicate end of file.
64 $40 1 FREQ	Beep count retain and count number of beeps requested of beep routine by cassette handler during open processing.
65 $41 1 SOUNDR	Noisy I/O flag when I/O is done buzzer sounds poke 0 and it won't buzz.
66 $42 1 CRITIC	Defines critical section (if non-zero) checked on NMI process after stage 1 processed.
67 $43 7 FMSZPO	Disk file manager zero page.
74 $4A 1 CKEY	Cassette boot request flag on powerup (coldstart). Start key checked, if pressed then CKEY is set.
75 $4B 1 CASSBT	Cassette boot flag.
76 $4C 1 DSTAT	Display status used by display handler.
77 $4D 1 ATRACT	Attract flag set to 0 by IRQ whenever a key is pressed. Incremented every 4 seconds by stage 1 Vblank. When value is 127 then value is set to $FE until attract mode is terminated.
78 $4E 1 DRKMSK	Dark attract mask=$FE when attract mode inactive.
79 $4F 1 COLRSH	Attract color shifter XOR'd with playfield colors. At stage 2 Vblank color registers are XOR'd with COLRSH and DRKMSK then sent to hardware color registers. When attract inactive COLRSH=0 and DRKMSK=$F6 reducing luminence 50% and COLRSH=RTCLOCK+1 affecting color change every 256/60=4.1 sec.
80 $50 1 TEMP	Used by display handler in moving data to and from screen.
81 $51 1 HOLD1	Same as [$50]. When BASIC in use these 2 locations called LOMEM and point to 256 byte buffer at end of OS. RAM used to tokenize one line of BASIC.
82 $52 1 LMARGN	Column of left margin of text screen, initialized to 2.
83 $53 1 RMARGN	Column of right margin of text screen initialized to 39. Margins are user alterable. Ignored in every mode but 0.
84 $54 1 ROWCRS	Display row number used in graphics screen and mode 0. Range 0 - 191. This location and COLCRS define the cursor location for the next data element to be read/written to main screen segment.
85 $55 2 COLCRS	Display column number used in graphics and mode 0 (lobyte). Range 0 - 319 (hibyte). Home position is 0,0 for both graphics and text.
87 $57 1 DINDEX	Display mode current screen mode obtained from low order 4 bits of most recent open AUXI byte.
88 $58 2 SAVMSC	Lowest address of display memory this location corresponds to the upper left corner of screen (lobyte).
90 $5A 1 OLDROW	These next 3 locations are updated from ROWCRS and COLCRS before every operation???
91 $5B 2 OLDCOL	These variables used only in draw and fill commands (lobyte).
93 $5D 1 OLDCHR	Retains value of character under visible text cursor. Used to restore character when cursor moves.
94 $5E 2 OLDADR	Retains memory address of current visible text cursor location. Used in conjunction with OLDCHR to restor character value when cursor moves lobyte.
96 $60 1 NEWROW	Point draw goes to.

continued on page 15

PILOT YOUR ATARI

by Ken Harms

PILOT is not just another computer language, it is designed to meet some of the needs of new programmers, educators, and children.

PILOT grew out of work by John Starkweather at the University of California at San Francisco back in 1972. He wanted a language which would make it easy to write tutorial programs for students, programs capable of recognizing responses other than the typical "1, 2, 3 multiple choice" style prevalent in current teaching programs. With PILOT, it is easy to ask, "Who was the first president of the United States?" and record and score answers such as "President Washington," "I believe it was G. Washington," "George Washington," "GEORGE WASHINGTON," "Washington". PILOT needs only three statements to accomplish this type of user interaction.

Dean Brown at Stanford Research Institute proved that teachers could understand PILOT, and students loved it. Since PILOT is word-oriented, as contrasted to BASIC's number orientation, it naturally fits the "riddle" and "tell-a-story" type of program which youngsters like. At the same time, Seymour Papert at MIT developed a new way to conceptualize and teach about geometry and shapes. This development was called "Turtle Graphics" and proved ideal for use in home computers. Atari wisely included a Turtle Graphics command language with the PILOT module.

The old "Cartesian Coordinate" system required commands like this:

> Start at position X=20 and Y=10. Draw a line to X=40 and Y=10; draw a line to X=40 and Y=30; draw a line to X=20 and Y=30; finally, draw a line to X=20 and Y=10.

Can you guess what figure this is? How big it is? Using Turtle Graphics the same pictures can be drawn like this:

> Do this 4 times: draw a line 20 space long, turn Right 90 degrees.

The box shape is more apparent and the commands are more readily understood. Only 14 or 15 commands represent the core of PILOT. All are only one or two characters long and easily remembered—a "J" is the "jump to" command. Anyone who is not a good typist will appreciate the wisdom of short commands. Short, easy to remember commands and Turtle Graphics combined with Atari's wonderful screen editor will make almost anyone's introduction to computing more pleasurable and rewarding.

Finally, PILOT programs become naturally organized around modules. This encourages a well-structured programming style. (Oh yes, PILOT includes full use of the Atari sound system—more on that in a later article.)

PILOT is available in two packages; one is just the language cartridge and users guide (about $90), the other is a well documented comprehensive package that I recommend (about $130). This package includes:

PILOT CARTRIDGE—(love those cartridges; little fingers can't destroy them)

STUDENT PILOT—a cleverly illustrated learner's manual for the new programmer.

PILOT PRIMER—an instruction manual for the experienced programmer.

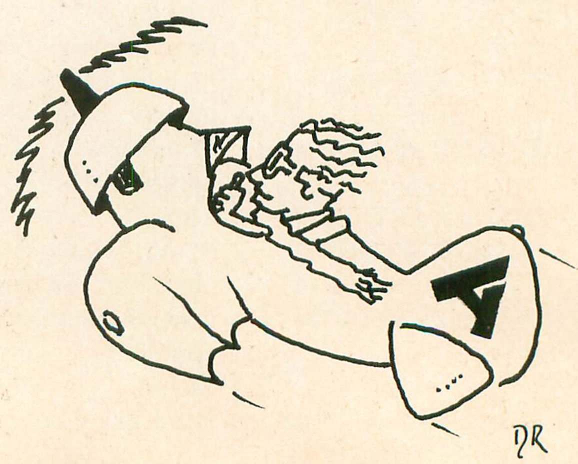
DEMONSTRATION TAPES—two cassettes showing language, color, graphics, and sound.

POCKET CHART—presents all commands in an easy-to-use format.

I like Atari's version of PILOT. There are still a few rough spots; not all syntax errors are caught, the manuals do not include indices, several commands are not explained in the manual, and a few typographical errors remain to confuse you. In spite of these few "start-up" problems Atari PILOT meets its "primary design goals"; it is "consistent and easy to learn . . . it allows reasonable access to the Atari system capabilities, but not at the user's expense."

We intend to help you get the most from PILOT. Watch for programming tips, warnings, and more help. Address your questions to:

K.W. Harms
c/o **ANTIC**
297 Missouri St.
San Francisco, CA 94107



GTIA

Over the coming months you'll be hearing more and more about GTIA, the new television interface chip. It is designed to replace the CTIA chip in all ATARIs currently on the market. In fact if you purchased your machine after January 1982, chances are you already have the GTIA installed. The rest of us will probably get a chance to purchase it as an upgrade.

The GTIA provides three additional graphics modes. These modes are variations of the normal Graphics 8 mode you are probably familiar with. The difference is that each pixel is 2 color clocks wide and 1 scan line high. This gives an effective resolution of 80 horizontally and 192 vertically. This is an unusual pixel size, wide and flat. The "pixel" or picture element is the smallest dot that can be manipulated on your television screen. The pixel size varies depending on the graphics mode chosen.

The three new modes are known as modes 9, 10 and 11.

Graphics 9 allows 16 different luminances of the same hue (color).

Graphics 10 allows 9 different colors and independent luminances.

Graphics 11 allows 16 different colors and one luminance.

The additional colors available with GTIA will make some interesting effects possible. We plan to fully explore this new graphics enhancement in future issues.

RESOURCES

For intermediate and advanced users the following references are recommended. These manuals are available from:

Customer Support
Attn: Julie Naughton
Atari Inc.
1312 Crossman Ave.
Sunnyvale, CA 94086
or phone 800-672-1430 in CA
800-538-8547 outside CA

Technical User's Notes
CO16555
This is an indispensible reference.
$29.95
CA residents add 6.5% tax
Outside CA add $3.50 shipping charge

ATARI 400/800 Operating System Source Listing
CO17893
Listing of 10K ROM excluding floating point routines.
$19.95
CA residents add 6.5% tax
Outside CA add $3.50 shipping

TAPE TOPICS

HELP FOR CASSETTE OWNERS

by Gary Phillips

The ATARI 400 or 800 with a cassette recorder is more useful than most users are aware. Many ATARI disk users have ignored the cassette unit as a cumbersome device at best. If your programming technique does not exploit the strengths and circumvent the weaknesses inherent in a cassette-only system then it is very cumbersome. When used with operating techniques like those we will be discussing in this department the cassette unit is a cost effective storage medium.

A TAPE OPERATING SYSTEM APPROACH

In the 1960s, large companies effectively used tape only systems for program and data storage and retrieval. A methodology called a Tape Operating System (TOS) was developed to simplify the programmers interaction with these devices.

A TOS was there to provide users with the ability to easily load, save, merge, and read or write program files to tape. With a few simple techniques these functions can be combined into a simple ATARI TOS.

The basic procedure described for the program recorder in the Atari documentation is to use one side of a cassette tape per program. This uses lots of tape and is slow and clumsy. Putting multiple programs on one tape by skipping forward to specific numbers on the tape counter seems to introduce as many problems as it solves.

A MULTI-PROGRAM "TAPE LOAD"

The key to more efficient use of the cassette lies with ability to merge two programs into one "tape load." This is done by numbering the two programs into different ranges of BASIC statement numbers, then merging them using the ENTER "C" command. Now a CSAVE will save the two programs together, and the next CLOAD will load both programs. Merging in a third and fourth program is done using the same method. Even a 16K ATARI will easily hold a dozen or more small-to-medium sized BASIC programs.

A MENU TO FIND PROGRAMS

One problem with having a number of different programs in memory at the same time is *how to find* and run the one you want. A simple solution to this problem is to include a "menu" program in the "tape load". The menu program contains the name and starting line number of each program in data statements. It displays a numbered list of programs on the screen, saving the starting numbers in an array by program number. It then invites the user to key in the number of the selected program. This number (REPLY in the sample listing) is used in the GOTO to go to the desired starting line number from the array of numbers. Menu numbers not in use are directed back to the menu by the starting line number of zero.

The menu system becomes a closed loop if every program in the tape load ends with a GOTO to the menu. I like to do this by ending all of the programs in a tape load with GOTO 0, and adding a line 0 GOTO 15000, if 15000 is the starting line number of the menu. This has two properties. A simple RUN will always display the menu. A program can be developed with easy-to-type low line numbers without interfering with the menu set-up.

MAKING MULTI-PROGRAM TAPE LOADS

This whole approach assumes you have a way to renumber BASIC programs. You can keep a renumbering program as part of all your tape loads, if you've got plenty of memory. Or, keep one in LIST "C" format on tape so you can write a new program on a tape by itself with LIST "C", specifying the line numbers the program uses, then clear memory with NEW, reload the program with ENTER "C", then merge in the renumbering program with ENTER "C". Once the new program is renumbered, resave it with LIST, load (CLOAD) the tape load you want it in and merge it in with ENTER "C". Now just put the name and starting line number in the list of data statements and you're in business.

continued

MULTI-PROGRAM TAPE LOADS, A FEW PROBLEMS

This technique will make life simpler but a few problems remain. Programs will be executed one after the other without an intervening END statement. This will lead to errors from multiple DIMENSION statements unless each program module begins with a CLR. Since BASIC considers all of the programs in the tape load as one continuous program the pointer to the current DATA statement is apt to get confused. This is remedied by including a RESTORE statement at the beginning of every program module. Another problem is that the same variable names may be reused in the separate program modules of a tape load. This will cause errors if the programs assume variables are set to zero or blank. This is corrected by initializing variables at the start of every module. Reusing variable names is useful to reduce program size. Remember there are only 128 available and it's often necessary to reuse names.

A more insidious problem occurs when a statement number is mistyped. This may omit the line from the program module you meant to change and clobber the line in another module. You can protect yourself against this by using separate working tapes for each program module and merging the working tapes to create a final tape load.

IN THE FUTURE

In future TAPE TOPICS we will discuss faster load techniques (600 baud too slow, how about 1200 or 2400 baud). We'll try to shorten the 20 second tape leader to just a couple of seconds. We will try to serve as a clearing house for tape problems and what to do about them. If you have questions, problems or suggestions send them to me c/o **ANTIC**.

See you tape users again in June!

Sample listing

```
0 GOTO 15000
15000 REM CODE TO SELECT A PROGRAM FROM
15005 REM A MULTI-PROGRAM TAPE LOAD
15010 CLR :DIM SNA(15):DIM NAME$(25)
15020 RESTORE (710):REM RESET DATA STMTS
15035 PRINT " "
15040 PRINT "NO. STMT #  PROGRAM NAME"
15050 FOR N=1 TO 8
15060 READ NAME$,STMTNO
15070 SNA(N)=STMTNO
15080 PRINT N;"   ";STMTNO;"    ";NAME$
15090 NEXT N
15100 PRINT "NO. OF DESIRED PROGRAM"
15110 PRINT "OR 0 TO EXIT TO 'READY'"
15120 TRAP 15120:INPUT REPLY
15130 IF REPLY<1 THEN END
15140 IF REPLY>8 THEN 15000
15150 GOTO SNA(REPLY)
15160 DATA SAMPLE PROGRAM A,18000
15170 DATA SAMPLE PROGRAM B,19000
15180 DATA SAMPLE PROGRAM C,20000
15190 DATA NOT DEFINED,15000
15200 DATA NOT DEFINED,15000
15210 DATA NOT DEFINED,15000
15220 DATA NOT DEFINED,15000
15230 DATA NOT DEFINED,15000
15240 DATA NOT DEFINED,15000
18000 REM SAMPLE PROGRAM A
18010 PRINT "SAMPLE PROGRAM A RUNNING"
18020 GOTO 0:REM RETURN TO MENU
19000 REM SAMPLE PROGRAM B
19010 PRINT "SAMPLE PROGRAM B RUNNING"
19020 GOTO 0:REM RETURN TO MENU
20000 REM SAMPLE PROGRAM C
20010 PRINT "SAMPLE PROGRAM C RUNNING"
20020 GOTO 0:REM RETURN TO MENU
```

MEMORY MAP *continued from page 11*

97	$61	2	NEWCOL	Column draw command uses.
99	$63	1	LOGCOL	Points at column in logical line. A logical line can contain up to 3 physical lines. This variable is used by display handler.
100	$64	2	ADRESS	Temporary storage holds contents of SAVMSC [$58]. and SAVMSC+1 [$59].??
102	$66	2	MLTTMP	OPNTMP first byte used in open as temp.??
104	$68	2	SAVADR	???
106	$6A	1	RAMTOP	RAM size defined by power on logic.
107	$6B	1	BUFCNT	Screen editor current logical line size.
108	$6C	2	BUFSTR	Editor low byte???
110	$6E	1	BITMSK	???
111	$6F	1	SHFAMT	Pixel justification???
112	$70	2	ROWAC	Accumulator control row point plotting and increment and decrement functions???
114	$72	2	COLAC	Controls column point plotting.

MEMORY MAP (Page 0) to be continued in next issue.

SOFTWARE REVIEWS

Eastern House Software
3239 Linda Drive
Winston-Salem, N.C. 27106
$49.95 from dealer or direct from publisher
Monkey Wrench (TM)

At last someone has put the right slot of the 800 to good use. This ROM-based product is designed to be installed in the right-hand slot of the 800 and used in conjunction with BASIC.

This product will provide the BASIC programmer with nine new commands: (A) auto line numbering, (D) delete range of lines, (M) change margins, (T) memory test, (R) renumber, (E) cursor movement without control key, ($) hex conversion, (#) decimal conversion, and a machine language monitor providing another 15 commands.

The BASIC commands must be used in immediate mode and must be preceded by the > prompt.The > symbol must appear in column one followed immediately by the command letter. Any necessary parameters are separated by a space.

The Machine Language Monitor is entered by the * command. The MLM prompt is a period (.). The 15 MLM commands are; (M) display memory range, (I) interrogate memory (similar to M but also displays ASCII equivalents), (R) display all 6502 register contents, (:) alter memory, (;) alter 6502 registers, (G) goto address and execute code, (X) exit MLM and return to DOS, (S) save memory to cassette, (L) load memory from cassette, (H') hunt for ASCII string, (H) hunt for hex string, (D) disassemble memory, (,) alter disassembled code, (B) calculate branch, and (A) enter ATARI DOS. Three normal MONKEY WRENCH commands are available; E, #, and $.

As good as this product is there are a couple of caveats. First, the cartridge uses address space hex 8000 to hex 9FFF. This will reduce your 48K machine to 32K, 8K used by the BASIC cartridge and 8K used by Monkey Wrench. Second, this product uses part of page 6 for variable storage which is sure to cause some conflict. One other important point is that care is needed when inserting the cartridge, it's very easy to put it in backwards. This is probably my only real critism since it would have been very easy to mark "front" on the circuit board.

I must give this product high marks. It is extremely useful, the little manual is excellent, and they've provided a second ROM socket for expansion. If you are a heavy BASIC user I recommend it. (Jim)

Santa Cruz Educational Software
5425 Jigger Dr.
Soquel, Ca. 95073
Tricky Tutorials (TM)

These programs shine a much needed light inside the ATARI. If you've had your hands on the O.S./Hardware manual, the one that costs 30 dollars and is guaranteed to keep you up nights, then you know we need better instruction. If you've looked with envy at the animation in Missile Command (TM), the smooth scrolling graphics window in Crawford's Eastern Front and the mixed text modes in File Manager (TM) 800, then look again at these tutorials. They make these techniques more accessible. There is a caveat, these programs are meant to instruct with the full attention and motivation of the user. If you're content to just watch, go to a movie, if you want to learn about the working internals of your ATARI and use the special features to design professional programs then load a TRICKY TUTORIAL (TM). It's almost impossible to use these products without getting some new insights. Each program is designed to give you immediate feedback using the most powerful tool at hand, your graphics display. The computer is used to provide the answer to your "I wonder what will happen when . . ." questions. Should you purchase these products be prepared to get involved, it's the only way.

There are currently six TRICKY TUTORIALS; 1) Display list, 2) Scrolling, 3) Page flipping, 4) Animation, 5) Player-Missiles, 6) Sound and Music.

Programs may be purchased individually or in a package of six. The six-pack comes in a nicely produced three-ring binder. All programs are available on tape and disk. All require 16K for cassette and 24K for the disk versions. The exception is the tutorial on Player-Missiles, #5, requiring 32K and costing $29.95. This was not available at the time this review was written.

I recommend these products; the price is right; the information necessary, and the value received high. (Jim)

Santa Cruz Educational Software * Tricky Tutorials * Santa Cruz Educational Software * Tricky Tutorials * Santa Cruz Educational Software * Tricky

ATARI (tm) GRAPHICS AND SOUND MADE EASY!

SANTA CRUZ EDUCATIONAL SOFTWARE HAS WRITTEN A SERIES OF AFFORDABLE PROGRAMS THAT DEMONSTRATES MANY OF THE SPECIAL "TRICKS" THAT ONLY THE ATARI 400/800 COMPUTER IS CAPABLE OF DOING. WE OFFER EVERYTHING FROM A PROGRAM THAT DOES FANTASTIC HIGH RESOLUTION GRAPHICS TO ONE THAT ACTUALLY DIALS YOUR PHONE.....

★★★

MASTER MEMORY MAP(tm) — This is really the key to using the ATARI'S capabilities. We start out by explaining how to PEEK and POKE values into memory so that even new programmers can use this. Then we give you over 15 pages of the memory locations that are the most useful. The information is condensed from both the ATARI'S Operating System Manual and various articles and programs. It is, of course, useful even for experienced programmers as a reference. Also, we highly suggest that dealers offer this Memory Map to customers who request to be told how to use the power of the machine. We guarantee it will answer many of the questions you have about the machine.

$6.95

TRICKY TUTORIALS(tm)

#1: DISPLAY LISTS — This program teaches you how to alter the program in the ATARI that controls the format of the screen. For example: when you say graphics 8 the machine responds with a large graphics 8 area at the top of the screen and a small text area at the bottom. Now, you will be able to mix the various modes on the screen at the same time. Just think how nice your programs could look with a mix of large and small text, and both high and low resolution graphics. this program has many examples plus does all of the difficult caluculations!

#2: HORIZONTAL/VERITCAL SCROLLING — The information you put on the screen, either graphics or text, can be moved up, down or sideways. This can make some nice effects. You could move only the text on the bottom half of the screen or perhaps create a map and then move smoothly over it by using the joystick.

#3: PAGE FLIPPING — Normally you have to redraw the screen every time you change the picture or text. Now you can learn how to have the computer draw the next page you want to see while you are still looking at the previous page, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your memory size and how complicated the picture, you could flip between many pages, thus allowing animation or other special effects with your text.

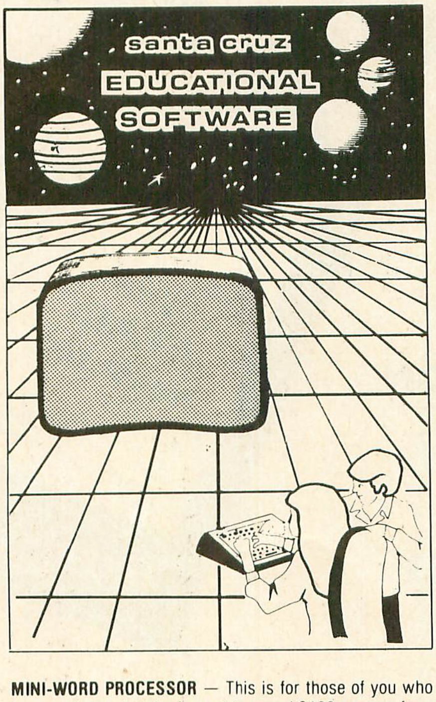
#4: BASICS OF ANIMATION — Shows you how to animate simple shapes using the PRINT and PLOT commands, and also has nice little PLAYER/MISSILE Graphics demo to learn. This would be an excellent way to start making your programs come alive on the screen. Recommended for new owners.

#5 PLAYER MISSILE GRAPHICS — This complex subject is demonstrated by starting with simple examples, and building up to a complete game and also an animated business chart on multiple pages! As always, the computer does most of the calculations. Requires 32K disk or tape and costs **$29.95**

#6: SOUND — From explaining how to create single notes, to demonstrating complex four channel sound effects, this newest tutorial is great. Even those experienced with ATARI's sound capabilities will find the menu of sound effects a needed reference that can be used whenever you are in the need of a special sound for your programs. Everyone will learn something new! Written by Jerry White.

Tricky Tutorials (except #5) require 16K memory for cassette orders and 24K for disk. The price is $19.95each. You may order 1,2,3, & 4 for $64.95.All six in a colorful binder cost $99.95.

THE GRAPHICS MACHINE!! — Turn your computer into an incredible graphics tool with advanced commands like circle, box, fill, polygon, line, help, etc. 3 colors in graphics 8 with instant text!!! Create colorful business charts or beautiful drawings and then save or retrieve them from disk in 5 SECONDS. YES, it's that fast. Needs all 48K, disk, and costs **$19.95**



MINI-WORD PROCESSOR — This is for those of you who have a printer, but don't want to spend $100 or more for a fancy word processor. It is suitable for simple editing of text, accepts most control characters for your printer, and text is stored on disk for easy retrieval. Holds 2½ typed pages at a time. Requires 32K, disk or tape. **$19.95**

BOB'S BUSINESS — 14 small business type programs for home or office, all chosen from a nice menu. Supports printed output. 169 sectors of output require 16K tape, or 32K disk. **$14.95**

KID'S #1 — Includes the following: 1) TREASURE — search for the lost treasure while trying to keep from falling into the sea. Nice graphics if you find it!; 2)DIALOGUE — talk back to the computer about four subjects; 3) MATH QUIZ — Nice musical and graphical rewards for good scores. Parents input the level of difficulty.

KID'S #2 — A spelling quiz, a "scrabble" type game, and a version of Touch with the computer giving all the directions! Both Kid's programs require 16K tape or 24K disk and cost **$14.95** each.

MINI-DATABASE/DIALER — This unique new program stores and edits up to 8 lines of information such as name, address, and phone numbers, or messages, inventories or anything you want. It has the usual sort, search, and print options, but it also has an unusual feature: If your files include phone numbers and you have a touch-tone phone, the program will DIAL THE PHONE NUMBERS FOR YOU! This is perfect for those who make a lot of calls like salesmen, teens, or those trying to get through to busy numbers (acts as an auto-redialer). It is also a lot of fun to use. Requires 16K cassette or 24K disk and costs **$24.95**

FONETONE — For those who only want to store name and phone numbers and have the dialer feature as above, we offer this reduced version. Same memory requirements, but only costs **$14.95.** Don't forget you must have a touch-tone phone.

PLAYER PIANO — Turns your keyboard into a mini-piano and more. Multiple menu options provide the ability to create your own songs, save or load data files using cassette or diskette, fix or change any of up to 400 notes in memory, and play all or part of a song. The screen displays the keyboard and indicates each key as it is played from a data file or the notes you type. You don't have to be a musician to enjoy this educational and entertaining program. Requires 24K cassette or 32K disk. **$14.95**

BOWLERS DATABASE — Provides the league bowler with the ability to record and retrieve bowling scores providing permanent records. The data may then be analyzed by the program and displayed or printed in summary or detail form. Data may be stored on cassette or diskette and updated quickly and efficiently. The program proivdes such information as highest and lowest scores by individual game, (first, second, and third games throughout the season), high and low series, current average, and more. The program listing and documentation provided are a tutorial on ATARI basic and record keeeping. Requires 16K for cassette or 24K for disk. **$14.95**

By the time you read this all computers (400/800) being produced should have the fabled GTIA chips included. ATARI service may upgrade older computers...call and ask (it's easy to do yourself). We have one and the improvements that graphics modes 9,10, and 11 offer are great!! To help you figure out what to do with the new modes a new Tricky Tutorial will be offered in March on Modes 9 to 11. Either give us a call or write around that-time.

★★★

SEE YOUR LOCAL DEALER
OR
ORDER FROM:

SANTA CRUZ EDUCATIONAL SOFTWARE
5425 Jigger Dr., Soquel, CA 95073
C.O.D. Orders call (408) 476-4901
***Includes $3.00 postage ($1.00 for Memory Map)**
***In California, include 6.5% tax**

Santa Cruz Educational Software * Tricky Tutorials * Santa Cruz Educational Software * Tricky Tutorials * Santa Cruz Educational Software * Tricky *

SUBSCRIBE

TO



- ☐ **BEGINNERS**
- ☐ **OLD TIMERS**
- ☐ **STARTING LINE**
- ☐ **LANGUAGES**
- ☐ **EDUCATION**
- ☐ **GAMES & DIVERSION**
- ☐ **PRODUCT REVIEWS**
- ☐ **BUSINESS**
- ☐ **PROGRAM LISTINGS**
- ☐ **PUBLIC DOMAIN SOFTWARE**

---

1 YEAR (6 ISSUES) $15

Enclose check or money order payable to:

ANTIC Publishing
297 Missouri Street
San Francisco, Ca. 94107

____
Name

____
Street

____
City, State Zip

Please check where appropriate:

I own:

☐ 800	☐ Cassette	___K Memory
☐ 400	☐ Interface	☐ Disk
☐ Modem	☐ Printer	

CHICKEN—A GREAT GAME

Why did the chicken cross the road?

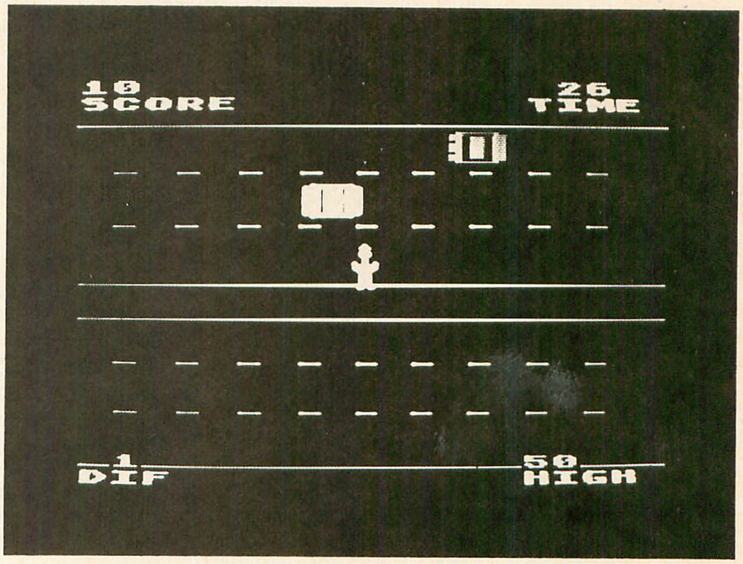
To lay an egg.

Actually, our chicken is trying to score points by getting safely across this busy highway. Each time he makes it adds to his score, but the cars go faster and faster. If he gets hit, the SPCA sends an ambulance and the cops slow the traffic down for a while.

This clever game can be yours for the copying, courtesy of Stan Ockers, who wrote it in BASIC and Assembly language and Mike Dunn, editor of ACE Newsletter (Eugene, Oregon), who printed it first, and gave us permission to pass it on to you.

The game is "in the public domain," meaning you can do anything you want with it except sell it.

We intend to bring you entertaining and useful programs that are in the public domain as a regular feature of our magazine.




```
0 REM **ACE NEWSLETTER 3662 VINE MAPLE DRIVE EUGENE, OR 97450**
1 REM **FEB 1982 ISSUE** $10 YEAR**
2 REM *******************
4 REM **   CHICKEN   **
6 REM ** S.O. 12-81 **
8 REM *******************
10 OPEN #1,4,0,"K:":DIM L$(20),S$(20),C$(20)
15 ? "INITIALIZING ..."
20 REM ** PAGE 6 ROUTINES AND DATA **
40 FOR I=1536 TO 1587:READ A:POKE I,A:NEXT I
41 REM ** VERTICAL BLANK ROUTINE **
42 DIM VB$(210):FOR I=1 TO 210:READ A:VB$(I)=CHR$(A):NEXT I
43 REM ** LOAD PLAYER ROUTINE **
45 DIM LD$(73):FOR I=1 TO 73:READ A:LD$(I)=CHR$(A):NEXT I
47 REM ** INSERT ADDRESS OF ROUT. IN PAGE 6 **
48 A=ADR(VB$):B=INT(A/256):POKE 1540,B:POKE 1538,A-256*B
50 DATA 104,160,52,162,6,169,7,76,92,228,104,160,98,162,228,169,7,76,92,228
52 DATA 120,120,120,120,30,57,81,105,15,15,15,15,0,0,0,0,52,53,54,55,2,2,3,4,12,0,0,0,15,11,11,11
54 REM ** LINE 56 - CHANGE 24 TO 56 TO SKIP ORTHO. **
55 REM ** LINE 56 - CHANGE 28 TO 34 FOR CONT. MOV. **
56 DATA 72,138,72,152,72,162,0,189,120,2,29,44,6,160,15,24,176,32,201,15,240,28,201,14,208,2,160,13,201,13
57 DATA 208,2,160,14,201,11,208,2,160,7,201,7,208,2,160,11,192,15,240,6,61,48,6,157,28,6,152,61,44,6,157,44,6
58 DATA 232,224,4,144,195
60 DATA 162,0,189,32,6,133,203,189,36,6,133,204,189,40,6,133,209,198,209,16,7,232,224,4
65 DATA 144,232,176,91,189,28,6,133,207
70 DATA 70,207,176,26,188,24,6,192,1,240,19,208,1,200,177
75 DATA 203,240,6,136,145,203,200,208,245,136,145,203,222,24,6,70,207,176,29,188,24,6,200,192,254,176,21
80 DATA 177,203,208,247,136,177,203,240,6,200,145,203,136,208,245,200,145,203,254,24,6,70,207,176,3,222,20
85 DATA 6,70,207,176,3,254,20,6,189,20,6,157,0,208
90 DATA 24,144,154,162,4,189,11,208,240,5,169,0,157,39,6,202,208,243,104,168,104,170,104,76,98,228
100 DATA 234, 234, 234, 104, 104, 104, 170, 189, 32, 6, 133, 186, 189, 36, 6, 133, 187, 104, 133, 213, 104, 133, 212
110 DATA 189, 24, 6, 133, 195, 169, 0, 168, 192, 255, 176, 35, 196, 195, 240, 5, 145, 186, 200, 208, 243, 162, 0, 161,
212, 240, 11
120 DATA 145, 186, 230, 212, 200, 192, 255, 176, 11, 208, 241, 169, 0, 145, 186, 200, 192, 255, 144, 249, 96, 234, 23
4
150 REM ** CAR COLOR DATA **
160 FOR I=1 TO 20:READ A:C$(I)=CHR$(A):NEXT I
170 DATA 24,60,218,68,90,186,70,150,54,232,74,168,88,154,21,252,200,76,228,28
190 REM ** DEFINE PM AREA - SINGLE LINE RESOL. **
200 A=PEEK(106)-16:POKE 54279,A:PM=256*A
205 REM ** PLAYER MISSILE POINTERS **
210 FOR I=4 TO 7:POKE 1568+I,A+I:NEXT I
```

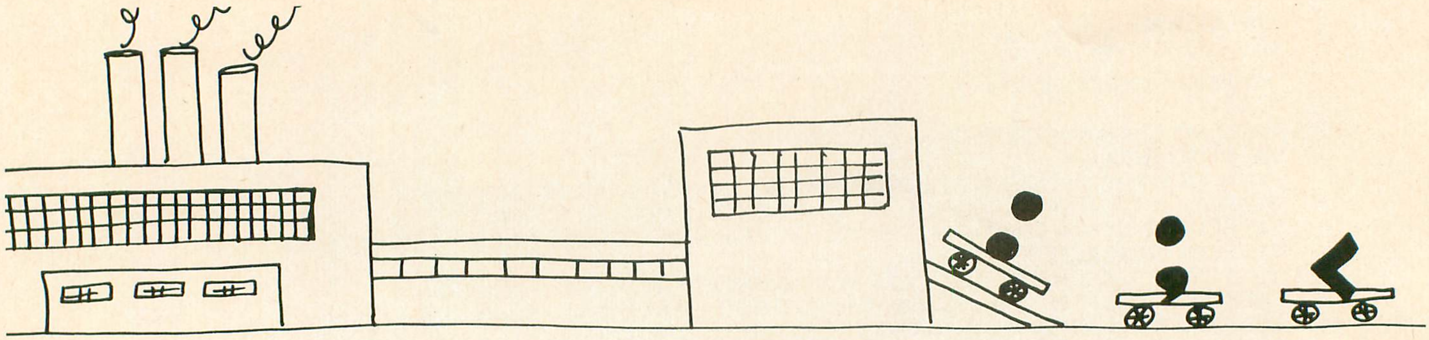
continued on next page

```
212 FOR I=1568 TO 1571:POKE I,0:NEXT I
218 REM ** DATA FOR PLAYER IMAGES **
220 FOR I=PM TO PM+121:READ A:POKE I,A:NEXT I
230 DATA 16,56,16,56,40,16,16,16,146,254,254,124,56,56,40,40,40,40,40,108,0
232 DATA 126,195,219,219,91,219,219,219,219,91,219,219,195,126,0
234 DATA 126,195,219,219,218,219,219,219,219,218,219,219,195,126,0
236 DATA 33,34,150,84,57,30,60,123,159,30,52,86,151,36,194,193,0
238 DATA 16,56,16,56,40,16,16,56,124,254,186,56,56,40,40,40,44,32,96,0
240 DATA 16,56,16,56,40,16,16,146,214,124,56,56,40,40,40,104,8,12,0
242 DATA 126,255,173,173,239,199,199,199,199,239,173,173,255,126,0
270 REM ** INIT. HORIZ. AND VERT. POS. **
280 RESTORE 282:FOR I=1556 TO 1563:READ A:POKE I,A:NEXT I
282 DATA 120,120,120,120,30,57,81,105
288 REM ** INIT COLORS **
290 DIF=3:BONUS=300:POKE 704,40:CP=0:FOR I=1 TO 3:POKE 704+I,ASC(C$(CP+I)):NEXT I:CP=3:BPOS=5
295 REM ** DRAW ROADS - SET PRIORITY **
300 GRAPHICS 17:FOR I=1 TO 20:L$(I)="_":NEXT I
305 FOR I=2 TO 20 STEP 2:S$(I)="-":S$(I-1)=" ":NEXT I
310 POSITION 0,2:? #6;L$:POSITION 0,11:? #6;L$:POSITION 0,13:? #6;L$:POSITION 0,22:? #6;L$
312 POSITION 0,5:? #6;S$:POSITION 0,8:? #6;S$:POSITION 0,16:? #6;S$:POSITION 0,19:? #6;S$:POKE 710,90
340 REM ** INIT. AND PRINT INFO. - RESET TIMER **
350 SCORE=50:POSITION 0,1:? #6;"score        time ":POSITION 0,23:? #6;"dif        high ";
360 POSITION 0,0:? #6;SCORE:POSITION 15,22:? #6;HIGH:POKE 77,0:POKE 19,0:POKE 20,0
365 REM ** INIT. PM GR. - FLAGS **
370 POKE 559,62:POKE 53277,3:I1=68:I2=88:FL=I1
375 REM ** LOAD PLAYERS - SET COLORS - PLAYER SIZES **
380 LD=ADR(LD$):A=USR(LD,0,PM):A=USR(LD,1,PM+21):A=USR(LD,2,PM+21):A=USR(LD,3,PM+21)
385 A=USR(1536):REM * INSERT VBI ROUTINE
390 POKE 53257,1:POKE 53258,1:POKE 53259,1:POKE 623,1
393 REM ** INIT. SPEEDS **
395 POKE 1576,2:FOR I=1577 TO 1579:POKE I,RND(0)*DIF+1:NEXT I
398 POSITION 1,22:? #6;DIF
400 REM ** IF CARS OFF SCREEN, CHANGE LANES **
410 IF PEEK(1557)<15 AND PEEK(1561)=57 THEN POKE 1561,193:A=USR(LD,1,PM+36):POKE 1585,7:GOSUB 1000:P
OKE 705,C
420 IF PEEK(1557)>240 AND PEEK(1561)=193 THEN POKE 1561,57:A=USR(LD,1,PM+21):POKE 1585,11:GOSUB 100
0:POKE 705,C
430 IF PEEK(1558)<15 AND PEEK(1562)=81 THEN POKE 1562,169:A=USR(LD,2,PM+36):POKE 1586,7:GOSUB 1000:P
OKE 706,C
440 IF PEEK(1558)>240 AND PEEK(1562)=169 THEN POKE 1562,81:A=USR(LD,2,PM+21):POKE 1586,11:GOSUB 100
0:POKE 706,C
450 IF PEEK(1559)<15 AND PEEK(1563)=105 THEN POKE 1563,145:A=USR(LD,3,PM+36):POKE 1587,7:GOSUB 1000
:POKE 707,C
460 IF PEEK(1559)>240 AND PEEK(1563)=145 THEN POKE 1563,105:A=USR(LD,3,PM+21):POKE 1587,11:GOSUB 10
00:POKE 707,C
465 REM ** PRINT TIME - CK. FOR TIME UP **
470 TIME=30-PEEK(19):POSITION 16,0:? #6;TIME;" ":IF TIME<=0 THEN 910
472 REM ** RESET SOU. - HORN ROUT. **
473 SOUND 0,0,0,0
475 IF RND(0)>0.5 THEN SOUND 1,0,0,0
480 IF RND(0)<0.05 THEN SOUND 1,7,12,10
482 REM ** CHICKEN STOMP **
485 P=PEEK(1564):IF P>15 OR P<13 THEN 500
490 IF P=15 THEN A=USR(LD,0,PM):GOTO 500
492 IF FL=I1 THEN FL=I2:SOUND 0,16,6,8:GOTO 496
494 IF FL=I2 THEN FL=I1:SOUND 0,22,6,8
495 REM ** CK. FOR REACHING BOTTOM **
496 A=USR(LD,0,PM+FL)
500 IF PEEK(1560)>230 THEN 810
505 REM ** CK. FOR COLLISION **
510 IF PEEK(53260)=0 THEN 410
515 REM ** SPLAT **
520 A=USR(LD,0,PM+51):FOR J=1 TO 3:SOUND 0,RND(0)*255,8,8:SOUND 1,RND(0)*255,8,8
525 POKE 704,PEEK(704)+8:FOR I=1 TO 30:NEXT I:NEXT J:SOUND 0,0,0,0:SOUND 1,0,0,0:POKE 704,40
527 REM ** DECREASE SCORE - CK FOR 0 **
530 SCORE=SCORE-20:POSITION 0,0:? #6;SCORE;"   ":IF SCORE<=0 THEN 702
533 REM ** MOVE UP - RELOAD BIRD - RESET COLL. **
535 POKE 1560,PEEK(1560)-24:A=USR(LD,0,PM):IF DIF>1 THEN DIF=DIF-1
590 POKE 53278,0:GOTO 395
700 REM ** SCORE = ZERO **
702 POSITION 3,2:? #6;"CHICKEN'S DEAD"
```

CHICKEN

continued from page 20

```
704 REM ** AMBULANCE **
705 C=1
706 IF PEEK(1576+C)=0 THEN C=C+1:GOTO 706
707 IF C>3 THEN C=1
713 POKE 1576+C,1:POKE 1560+C,PEEK(1560):POKE 1556+C,220:A=USR(LD,C,PM+107)
715 FOR J=1 TO 6:FOR P=60 TO 40 STEP -2:SOUND 0,P,10,8:FOR I=1 TO 6:NEXT I
716 NEXT P:FOR P=40 TO 60 STEP 2:SOUND 0,P,10,8:FOR I=1 TO 6:NEXT I:NEXT P:NEXT J:SOUND 0,0,0,0
718 REM ** NEW HIGH SCORE? **
720 A=USR(1546):IF SCORE>HIGH THEN HIGH=SCORE:POSITION 15,22:? #6;HIGH
730 POSITION 2,6:? #6;"press FIRE button":POSITION 4,7:? #6;"to play again"
732 FOR I=53248 TO 53251:POKE I,0:NEXT I:SOUND 0,0,0,0:SOUND 1,0,0,0
735 REM ** WAIT FOR BUTTON **
740 IF STRIG(0)=1 THEN 740
745 REM ** PM GRAPHICS OFF **
750 POKE 53278,0:POKE 53277,0:A=USR(1546):GOTO 280
800 REM ** BK TO TOP - STOP MOVEMENT OF BIRD **
810 POKE 1560,30:A=USR(LD,0,PM):POKE 1576,0
815 REM ** SIGNAL AND INCREMENT SCORE **
820 FOR I=1 TO 5:FOR J=10 TO 5 STEP -1:SOUND 0,J,14,8:SOUND 1,J,2,8:NEXT J:SOUND 0,0,0,0:SOUND 1,0,0,0
825 A=USR(LD,0,PM+68):FOR J=1 TO RND(0)*30:NEXT J:A=USR(LD,0,PM+88)
830 SCORE=SCORE+DIF*2:POSITION 0,0:? #6;SCORE;"  ":NEXT I
840 IF DIF<9 THEN DIF=DIF+1
841 REM ** CK FOR BONUS **
842 IF SCORE<BONUS THEN 850
844 SOUND 0,25,10,10:BONUS=BONUS+300:P=PEEK(19):IF P<11 THEN POKE 19,0:GOTO 848
846 POKE 19,P-10
848 POSITION BPOS,13:? #6;"*":BPOS=BPOS+1
850 GOTO 390
900 REM ** TIME'S UP ROUTINE **
910 POSITION 5,2:? #6;"TIME'S UP"
920 GOTO 720
990 REM ** CHANGE CAR COLOR SUB.**
1000 CP=CP+1:IF CP=20 THEN CP=1
1010 C=ASC(C$(CP)):RETURN
```



FORTH FACTORY

by Bob Gonsalves

In this premier issue of **ANTIC** I would like to introduce a facility I believe many programmers using the Atari 800/400 will find very useful. The tiny MULTI-TASKING kernel described here allows one to easily create machine language routines that run as part of the vertical blanking interrupt servicing routines. These routines may perform such diverse tasks as moving a player/missile image, blinking the cursor, fine scrolling of the bit map, and various musical applications.

The run-time kernel of the multi-tasking system is contained on screen 36 and 37. There are three variables that are accessed by the routine. TASK# indicates which of several tasks will execute during the current vertical blanking interval. It is used to index into a table, called TASK-TABLE. The table is 16 bytes long, and holds the entry points for up to 8 separate machine code tasks. TASK# indicates which of these entries is to be executed; the entry point address is transferred from the indexed location in the task table, to JUMPER and an indirect jump is made through JUMPER.

The routines of scr#37 perform the necessary run-time operations. The master vblank routine is called NEW-VBLANK. It in turn calls two other routines. The first, WHOSE-TURN, uses the value of TASK# to find the appropriate slot in the task table, which contains the address of a subroutine. Notice that if the most significant byte of the entry in the table is set to 0, we can assume that the slot really does not contain a valid address. Otherwise, the entry point address held at that slot is transferred to JUMPER and we vector through JUMPER to the actual subroutine that we want to execute.

When we return from that subroutine, we next execute NEXT-TASK, which simply bumps TASK# by 1, rolling over to 0 if it is greater than three. We then leave the NEW-VBLANK routine by jumping to the code in the Atari Operating System that updates various hardware registers from their shadow registers.

Screen 35 establishes the equates for interfacing with the OS. SET-VBLANKD provides the linkage between Forth and Atari supplied routine for changing the vertical blanking vectors and countdown timers. To install a new vertical blanking vector, you simply push the new routine address onto the stack before call SET-VBLANKD.

Screens 38 and 39 provide us with a means of adding and deleting entries from the task-table. The routines MTOFF and MTON disable and enable the multi-tasking respectively, by changing the deferred vertical blanking vector, using our SET-VBLANKD routine. SNEAK is a way of ensuring that operations that change some variable read by the multi-tasking do so only when the multi-tasking is disabled. SCAN-TASKS is used to find an entry in the task table. It can be used to either find an available slot (an entry in the table=0) or to find the slot holding the address of a task routine. INSTALL and REMOVE will perform these actions for us. They can be used in the form

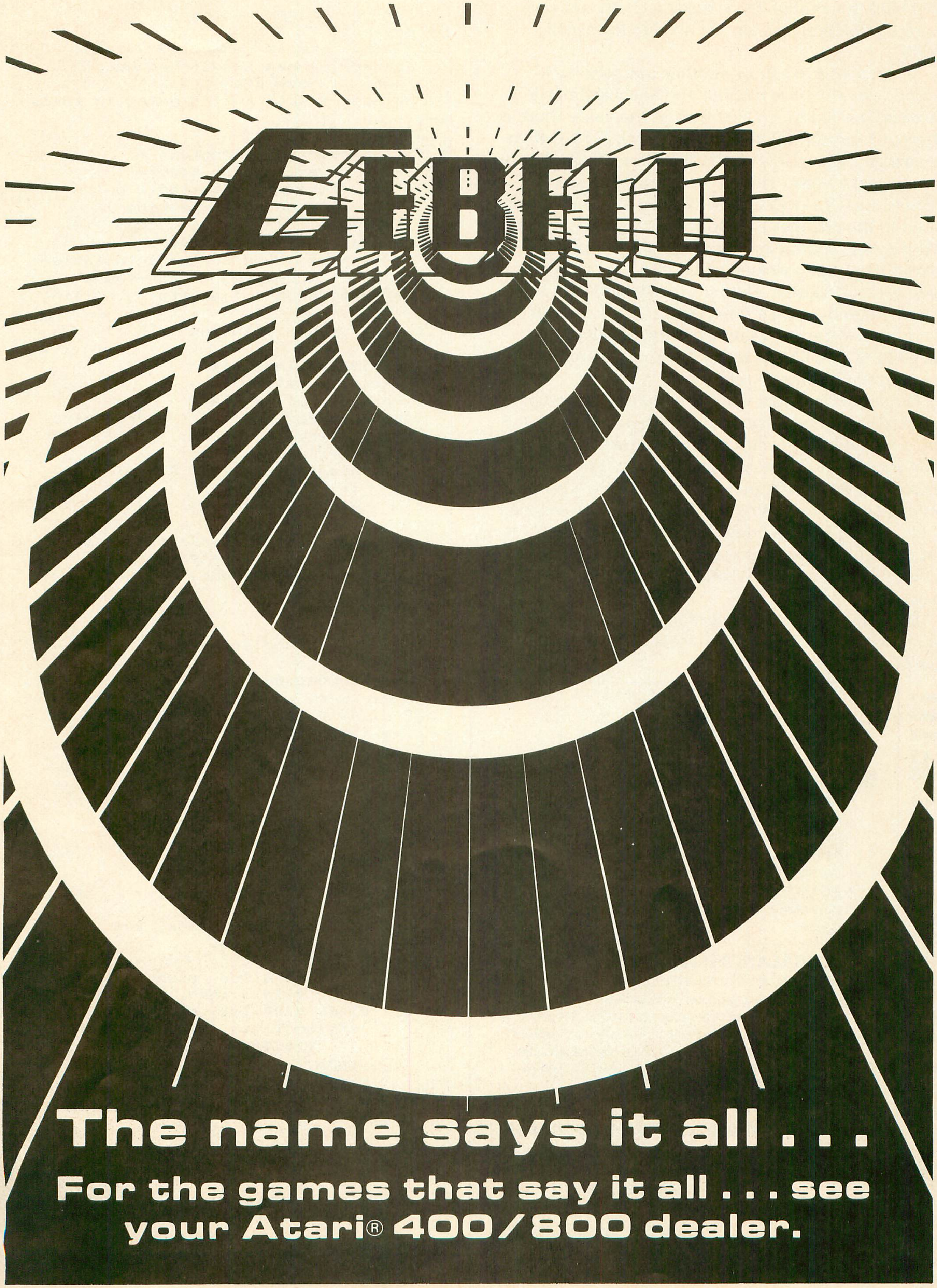
INSTALL BLINK (for example)
 or
REMOVE BLINK

where BLINK is one of the tasks we have defined.

TASK: and ;TASK are defining type words which will allow us to create machine code routines, which are subsequently installed in the task table. An important feature of the compilation security of these words is that the routines *are not* installed into the task table if some sort of structural error was generated by assembling them.

Finally, we can show some screens, that, when compiled, will install some tasks into the multi-tasking system. Once this is done, the routines operate *independently* of what you are doing as you program in the high level language. They have been integrated into the interrupt structure of the Atari Operating System. Screen 41 provides a task which will period-

continued on page 24


GEBELLI
The name says it all . . .
For the games that say it all . . . see your Atari® 400/800 dealer.

FORTH FACTORY *continued from page 22*

ically blink the cursor. Once installed, you can vary the rate of blinking by writing a value to variable RATE.

Screens 42 & 43 show another amusing background task. Here we monitor the keyboard hardware register (address D209 hex). If the value there is different from the value contained at OLD.KEY, then we change the variable SOUNDING to indicate that we want to make noise. The current key value then becomes the OLD.KEY value. The key value is also stored to the channel 1 audio frequency register. The next time the routine executes, we store a value, derived from the envelope table, into the channel 1 audio control register. The index variable INTO is decremented by 1 each time, until it is less than 0 (we've loaded the last value from the envelope). At this point SOUNDING is set to 0 and we resume checking the hardware register.

Other routines that various people have designed for BASIC could be integrated into this system. These would include the joystick controlled text cursor, and using some keys (space bar, Start key, etc) to temporarily halt program execution. Those who are contemplating purchasing the Forth system produced by Pink Noise Studios might be interested to know that a complete player graphics system is supported using this multi-tasking system to refresh the player images.

Should you have any questions regarding this column or pns fig FORTH—address them to me in care of **ANTIC.**

```
( 39 new-vblank process                          )
 decimal
: TASK:  create [compile] assembler
  assembler mem ( switches now  )
  here  ( save pfa )
  0 scan-tasks  ( find entry in table )
  !csp
  mtoff ;
: ;TASK current @ context !
  ?csp  ( everything secure )
  !  ( store pfa into table entry )
  mton  ; ( do multi-tasking )
: INSTALL
  [compile] ' 0 scan-tasks
   sneak  ; ( the change into table )
: REMOVE  [compile] ' scan-tasks
  0 swap sneak ; ( delete by nulling )

( 41 trying a blinking task                      )
 hex
0 variable time-left
0 variable rate
0 variable blank-state
d401 constant chactl  ( char control )
 task: blink
 time-left  lda, 0=
  if, blank-state lda, 0= NOT
      if, 0 # lda,
      else, 1 # lda,
      then,
     chactl sta, blank-state sta,
     rate lda, time-left sta,
  else, time-left dec,
  then, rts,
 ;task

( 42 sound handler example                       )
 hex
label envelope
a0 c, a1 c, a2 c, 00 c, a5 c, a6 c,
a8 c, 00 c, ab c, ac c, 00 c, af c,
c variable into
0 variable sounding
0 variable old.key
 decimal

( 43 sound handler example                       )
 hex
0 voice
 task: key.noise
  sounding lda, 0= not
     if, into ldx,
         envelope ,x lda, audc sta,
         volume sta,
         dex, 0<
        if, 0 # lda, sounding sta,
         c # ldx, then, into stx,
     else, d209 lda, old.key cmp, 0= not
      if, audf sta, old.key sta,
          1 # lda, sounding sta,
      then,
     then, rts,
 ;task
 decimal
```

```
( 35 new vblank vectors                    )
 HEX
e45c  constant SETVBV
e460  constant VBLANKI
e463  constant VBLANKX
224 @ constant VBLANKD ( OS dependant )
code Set-Vblankd ( addr is on stack )
  pha, tya, pha, xsave stx,
  bot lda, pha,  ( IS ON STACK )
  bot 1+ lda, tax, pla, tay,
  7 # lda, setvbv jsr,
  pla, tay, pla, xsave ldx,
  pop jmp,
end-code
 decimal
( the 7 # lda, would be changed to 6 )
( for the immediate vblankd vector)
( and to 1-5 for timers 1-5 )

( 36 setting up task table                 )
00 variable TASK#
00 variable JUMPER ( indirect jump )
 label TASK-TABLE
00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 ,

( 37 setting up task table                 )
code WHOSE-TURN
   task# lda, .a asl, tax,
   task-table 1+ ,x lda, 0= not
   if, jumper 1+ sta, ( install pointer)
        task-table ,x lda, jumper sta,
        jumper ) jmp,
   then, rts,
end-code
code NEXT-TASK  clc, task# lda,
   1 # adc, 03 # and, task# sta, rts,
end-code
( 4 entries in table are checked )
code NEW-VBLANK
   ' whose-turn jsr,
   ' next-task  jsr,
  vblankd       jmp, end-code
decimal

( 38 new-vblank process                    )
 decimal
: SCAN-TASKS ( match val)
   -1 swap 32 0 ( flag for 0<)
   do i task-table + dup ( addr in tabl)
     @ 3 pick =  ( match to given val )
     if rot drop  ( loose 0< flag)
        swap leave
     else drop  ( addr or flag )
     then
    2 +loop drop  ( given val )
    dup 0< 44 ?error  ;
( task table in lower 32K of memory !! )
: MTOFF  vblankd set-vblankd  ;
( disable and enable multi-tasking         )
: MTON  ' new-vblank set-vblankd  ;
: SNEAK  mtoff !  mton  ;
( synchronized storage                     )
```

An essential book for beginner and intermediate Forth programmers is:

Starting Forth
by Leo Brodie
Prentice-Hall
$15.95 softcover

The FORTH language is commercially available for the ATARI from three sources.

1. **Mountain View Press**
   P.O. Box 4656
   Mountain View, CA 94040
   Publishes "pns" Fig Forth

2. **ATARI Program Exchange (APX)**
   P.O. Box 427
   155 Moffett Park Drive
   Sunnyvale, CA 94086
   Publishes Fig Forth

3. **Quality Software**
   6660 Reseda Blvd.
   Suite 105
   Reseda, CA 91335
   Publishes Fig Forth

If you want to find out what all the excitement is about get one of the Forth implementations listed above and this book. Oh, and read **FORTH FACTORY**. Bob Gonsalves writes this column for us and is the implementor of the version available from Mountain View Press. Watch for a comparison of these three products in a future issue.

ASSEMBLER/EDITOR CARTRIDGE
TRICKS OF THE TRADE

by James Capparell

Do you often wonder how some of the incredible graphics were achieved in your favorite program. Looking at someone else's program is an excellent method of self-instruction. The problem is that often we only have the assembled (object) code in our possession. The L option of the A/E cartridge makes it possible to recreate source code from the object code. This is a useful feature. The catch is the inability to save the output of the L function to disk. Follow these simple instructions and you will be able to save the disassembled output to disk.

1. Load the object code to be disassembled. Unnecessary if you are dealing with ROM code.
2. Note the first memory address (hex) where the disassembly process is to begin.
3. Enter the debug mode of A/E by typing BUG and pressing return.
4. Start the disassembly process at the hex address noted in step 2. Enter the L command followed by the hex address and press return. This command disassembles 20 lines of code and displays the results on your T.V. (IMPORTANT—Note the last address printed on your screen).
5. Move the cursor to first line on screen using control and arrow keys and insert 2 blank lines using the shift and insert keys.
6. Return to the Edit mode by typing X and pressing return. The screen will still display the dis-assembled output.
7. Again move the cursor to the first line on the screen and enter Auto line numbering mode by typing NUM, press return. This will cause the cursor to print out the number 10 if this is the first time the auto numbering command has been executed or the next number incremented by 10. It will also place the number and the cursor at the start of the first line of disassembled code.
8. Using the control and delete key delete all characters upto two spaces prior to the assembly language mnemonic. The two spaces are neces-sary to conform to A/E spacing requirements.
9. Press return to enter this line of code into the source file. This will also generate the next line number incremented by 10. The cursor will appear at the beginning of the next line.
10. Continue with all 20 lines. When done, if you have more code to disassemble return to step 3. This can go on indefinitely or until source buffers are full at which time you will need to save the source file. The auto line numbering remembers the last line number and will correctly generate the proper sequences between 20 line chunks. This technique can be useful when you're trying to decipher code. Labels can be added in place of addresses in the operand fields and the corresponding label fields. This entire process is called unassembling object code. It can be very useful when used on small subroutines.

Note: *This same process also works on EDASM, Optimized Systems Software's assembler/editor.*

WHAT'S IN A NAME
ANTIC

Many old timers know that ANTIC is the name of just one of the large scale integrated (LSI) circuits inside the ATARI. This chip functions as a graphics controller and allows the ATARI to perform graphics better than almost any machine in it's price range and many costing much more. We liked the name ANTIC because of it's connotation of playfulness and it's direct connection with our machines. Our reference on the cover to "resource"means that we want to be a source of supply, support, or aid. We intend to live up to our name and have registered it as our trademark. Watch for software under our label.

MORE ON NAMES

Many of you probably are not aware of what the term ATARI means. The word itself originates from the ancient game GO. The term is used to warn one's opponent that capture of territory is imminent. It's a polite way of saying "attack." Since I am a GO player, I've always appreciated the name given to my favorite computer. (JC)

ASSEMBLY LANGUAGE

by Matt Loveless

The following assembly language program will allow a BASIC program the capability of selectively (masking) ignoring any key(s) on the keyboard. This function is accomplished by first using the BASIC program, SETUP, to choose those keys which are to be enabled or masked. The output of program SETUP is a file containing string data and machine code. This file may be appended to your own program as a subroutine and once called the desired masking effect will be enabled.

This routine puts to good use locations $208,$209, called VKEYBD. VKEYBD is accessed whenever a key is pressed on the keyboard. Normally VKEYBD contains $FFBE, the start of the system's keyboard interrupt service routine. Our setup routine BEGIN (see listing) puts $620 the start of our own keyboard interrupt routine in locations $208,$209. Now whenever a key is pressed our routine is executed which either ignores a key by simply performing an RTI (return from interrupt) instruction or jumps to the normal system routine at $FFBE and allows the key to be accepted.

To use the BASIC program, SETUP, type it in and save it. When SETUP is executed it will ask whether keys should be Enabled(E) or Masked(M). When in mode E all keys are initially masked (disabled) and you must selectively re-enable each key. If M mode is chosen then all keys are initially enabled and you must selectively mask (disable) each key.

In the following example all computer output is printed in bold letters. *See page 30 for* SETUP.BAS *listing.*

Listing 1

```
0000           10  .TITLE"KEY MASK 1.0"
               20 ;*******************************
               30 ;***        KEY MASK 1.0      ***
               40 ;***(c) 1982 by, Matt Loveless***
               50 ;***  written especially for  ***
               60 ;***       ANTIC magazine     ***
               70 ;*******************************
               80 ;
               90 ;*** EQUATES ***
               0100 ;
D209           0110 KBCODE =    $D209       ; Key board code
0208           0120 VKEYBD =    $0208       ; Keyboard IRQ ve
00CB           0130 TABLE  =    $CB         ; Free zero page locations
               0140 ;
               0150 ;===================================================
               0160 ;  INIT ROUTINE:  Puts my keyboard handler online
               0170 ;===================================================
0000           0180        *=   $0600       ; Page 6
0600 78        0190 BEGIN  SEI              ; Disable IRQ's
0601 AD0802    0200        LDA  VKEYBD      ; Point the keyboard vector to my
0604 8D3006    0210        STA  JMPLOC+1    ; routine, and set my routine's JMP
0607 AD0902    0220        LDA  VKEYBD+1    ; instruction to point to the OS's
060A 8D3106    0230        STA  JMPLOC+2    ; routine
060D A920      0240        LDA  #MYRTN&255 ; Lo byte of my routine
060F 8D0802    0250        STA  VKEYBD
0612 A906      0260        LDA  #MYRTN/256 ; Hi byte of my routine
0614 8D0902    0270        STA  VKEYBD+1
0617 68        0280        PLA              ; Remove USR amount byte
0618 68        0290        PLA              ; Get hi byte of the mask string
0619 85CC      0300        STA  TABLE+1
061B 68        0310        PLA              ; Get lo byte
061C 85CB      0320        STA  TABLE
061E 58        0330        CLI              ; Re-Enable IRQ's !
061F 60        0340        RTS              ; Return to BASIC
               0350 ;==================================================
               0360 ;THE NEW KEYBOARD INTERRUPT ROUTINE
               0370 ;==================================================
0620 98        0380 MYRTN  TYA              ; Keyboard IRQ vector points here
0621 48        0390        PHA              ; Save Y-register
0622 AC09D2    0400        LDY  KBCODE      ; Get the key code
0625 B1CB      0410        LDA  (TABLE),Y   ; and use it to index into the string
0627 D004      0420        BNE  GONORM      ; Is it masked out?
0629 68        0430        PLA              ; YES - then ignore key and
062A A8        0440        TAY              ; restore registers
062B 68        0450        PLA
062C 40        0460        RTI              ; Exit the keyboard interrupt
               0470 ;
062D 68        0480 GONORM PLA              ; Restore Y-register
062E A8        0490        TAY
062F 4C0000    0500 JMPLOC JMP  $0000       ; Go to normal system keyboard routine
0632           0510        .END
```

ASSEMBLY LANGUAGE *continued*

Would you like to ENABLE keys(E) or MASK keys(M)?
E(return)

—ENABLE MODE—
ALL KEYS ARE DISABLED.
PRESS KEY(s) YOU WANT TO ENABLE.
ABC(return)
press START key

WHICH LINE NUMBER TO START SUBROUTINE ON?
2500(return)

ENTER OUTPUT FILE NAME, INCLUDE DEVICE.
D:TEST.BAS(return)
NOTE: Subroutine will be written to disk at this point.

To test the TEST.BAS program.
NEW(return)
ENTER"D:TEST.BAS"(return)
RUN(return)
NOTE: Only key AB and C will be recognized all other keys are masked (disabled).

SUBSCRIBE TO ANTIC

See page 18

ASSEMBLY LANGUAGE *continued*

Listing 2

```
5 REM ** SETUP.BAS **
8 REM ** written for ANTIC by Matt Loveless
10 DIM KEY$(260),WRT$(1000),A$(1),FILE$(20)
15 ? "}"
20 ? "Would you like to ENABLE keys (E)":? "or MASK
   keys (M)";
25 INPUT A$
30 IF A$="E" THEN GOTO 100
40 IF A$="M" THEN GOTO 200
50 ? :? :GOTO 20
100 REM ---- ENABLE KEYS ---
110 REM ALL KEYS START OUT DISABLED
120 KEY$(1)=CHR$(0):KEY$(260)=CHR$(0):KEY$(2)=KEY$
130 GRAPHICS 0:SETCOLOR 2,0,0:? :?
131 ? "        ---- ENABLE MODE ----
132 ? "       ALL KEYS ARE DISABLED"
135 POKE 764,255:REM CLEAR CHARACTER BUFFER
140 POKE 752,1:? :? :? "   PRESS KEY(S) YOU WANT TO
    ENABLE."
150 POSITION 2,21:? "   press [START] after all keys"
155 ? "          are entered."
160 P=PEEK(764):IF P<>255 THEN 180
170 IF PEEK(53279)<>7 THEN 1000
175 GOTO 160
180 P=P+1:KEY$(P,P)=CHR$(1)
190 POKE 764,255:GOTO 160
200 REM ---- DISABLE MODE ----
210 REM ALL KEYS START OUT ENABLED
220 KEY$(1)=CHR$(1):KEY$(260)=CHR$(1):KEY$(2)=KEY$
230 GRAPHICS 0:SETCOLOR 2,0,0:? :?
231 ? "       ---- DISABLE MODE ---        "
232 ? "       ALL KEYS ARE ENABLED"
235 POKE 764,255:REM CLEAR CHARACTER BUFFER
240 POKE 752,1:? :? :? "   PRESS KEY(S) YOU WANT TO
    DISABLE."
250 POSITION 2,21:? "   press [START] after all keys"
255 ? "          are entered."
260 P=PEEK(764):IF P<>255 THEN 280
270 IF PEEK(53279)<>7 THEN 1000
275 GOTO 260
280 P=P+1:KEY$(P,P)=CHR$(0)
290 POKE 764,255:GOTO 260
1000 REM ---- CREATE STRING ----
1010 GRAPHICS 0:SETCOLOR 2,8,2:? :?
1015 ? "    WHICH LINE NUMBER TO STAR"
1016 ? "         SUBROUTINE ON";
1020 TRAP 1020:INPUT AMT
1025 ? :? :? "ENTER OUTPUT FILE NAME, INCLUDE DEVICE";
1026 INPUT FILE$
1028 OPEN #1,8,0,FILE$
1030 WRT$=STR$(AMT)
1032 WRT$(LEN(WRT$)+1)="DIM KEY$(257):KEY$(1,50)="
1040 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1050 WRT$(LEN(WRT$)+1)=KEY$(1,50)
1060 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1070 PRINT #1;WRT$
1080 AMT=AMT+10
1090 WRT$=STR$(AMT):WRT$(LEN(WRT$)+1)="KEY$(51,120)="
1100 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1110 WRT$(LEN(WRT$)+1)=KEY$(51,120)
1120 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1130 PRINT #1;WRT$
1140 AMT=AMT+10
1150 WRT$=STR$(AMT):WRT$(LEN(WRT$)+1)="KEY$(121,200)="
1160 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1170 WRT$(LEN(WRT$)+1)=KEY$(121,200)
1180 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1190 PRINT #1;WRT$
1200 AMT=AMT+10
1210 WRT$=STR$(AMT):WRT$(LEN(WRT$)+1)="KEY$(201,257)="
1220 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1230 WRT$(LEN(WRT$)+1)=KEY$(201,257)
1240 WRT$(LEN(WRT$)+1)=CHR$(34):REM QUOTATION MARK
1250 PRINT #1;WRT$
1260 AMT=AMT+10
1270 WRT$=STR$(AMT)
```

ASSEMBLY LANGUAGE
continued on next page

```
1280 WRT$(LEN(WRT$)+1)=" DATA 120,173,8,2,141,48,6,173,9,2,141,49,
     6,169,32,141,08,02,169,6,141,9,2"
1290 PRINT #1;WRT$
1300 AMT=AMT+10!WRT$=STR$(AMT)
1310 WRT$(LEN(WRT$)+1)=" DATA 104,104,133,204,104,133,203,88,96,
     152,72,172,9,210,177,203,208,4,104"
1320 PRINT #1;WRT$
1325 AMT=AMT+10!WRT$=STR$(AMT)
1330 WRT$(LEN(WRT$)+1)=" DATA 168,104,64,104,168,76,0,0"
1340 PRINT #1;WRT$
1350 AMT=AMT+10!WRT$=STR$(AMT)
1360 WRT$(LEN(WRT$)+1)="FOR ZQZ=1536 TO 1585!READ
     ZQX!POKE ZQZ,ZQX!NEXT ZQZ"
1370 PRINT #1;WRT$
1380 AMT=AMT+10!WRT$=STR$(AMT)
1390 WRT$(LEN(WRT$)+1)=" X=USR(1536,ADR(KEY$)): REM
     INITIALIZE"
1400 PRINT #1;WRT$
1410 CLOSE #1
1420 END
```

ASSEMBLER LANGUAGE

The following products are recommended.

The ATARI Assembler
by Don & Kurt Inman
Reston Publishing Company
$14.95 hardbound.
Suitable for beginners. See review in this issue.

6502 Assembly Language Programming
by Lance Leventhal
Osborne/McGraw Hill, Inc.
$15.95
A necessary reference text. Over the heads of beginners, but an inevitable purchase.

Assembler Editor language cartridge
Atari Inc.
$54.95
Most suitable for beginners. NOTE: EADASM from O.S.S. is very similar. For experienced Assembly language users there are better packages, watch for our recommendations in coming issues.

USER NOTES

Did you buy your computer after January 1982? If so, look for a "G" sticker on your console carton. This means you have the new GTIA chip (see page 13). Also look for a yellow "C" sticker and a blue "DS" sticker on your disk drive carton. These two stickers indicate that you have the new fast format ROM C chip and a data separator circuit. With with the addition of these electronic parts your disk drive will load programs faster and be more reliable. THANKS ATARI!!

LOOKING AT BOOKS

COMPUTE!'s First Book of ATARI
Published by *COMPUTE!* Books
Small System Services, Inc.
184 pages—$12.95

by DeWitt Robbeloth

This book is a collection of articles from *COMPUTE!* Magazine. All of the material is about the ATARI computer and its uses. That's very handy material if you are an ATARI owner. The biggest problem with the book is that the topics are scattershot, reflecting the interests of the many writers rather than the needs of the reader. Still, it is likely that sooner or later an ATARI owner will have an interest in most of the information in this book, and once you have the book, there it will be. I bought it for this promise alone.

Many of the articles are directly and immediately useful, for the beginner as well as the experienced user. Since these articles did appear in *COMPUTE!* at some time during the past few years, it is fair to assume that they met a relevance challenge from the magazine's editors when published, and the fact that they made it through another editorial screening is a good recommendation for their quality.

There are 33 articles by 24 different authors several of them well known in the micro world. Their work has been organized generally around several key interests: the BASIC language; graphics; programming techniques; and peripherals. The book is not a primer in any of these areas, nor is it a substitute for diligent study of the ATARI manuals. Each article stands alone, and if it is too advanced, the next one may well be easier. Almost all of the articles need to be studied to yield any real benefit. Many include program listings for user copying or saving. Most of the listings are photocopied from printouts, so you can assume they are accurate.

This book clearly tells you a lot about the ATARI that you couldn't find out from authorized sources and could only discover by chance on your own. In that respect, these articles are like war stories, swapped behind the lines during R&R. One guy says that creating 3-D effects with color graphics is a snap. (Oh, yeah!) Another describes the structure of BASIC inch by inch, as if it were an impregnable fort. Someone else demystifies a scary function. (I guess POKE ain't so hard.)

The best recommendation I can give for this book is that I bought it. *COMPUTE!* has a good reputation for supporting ATARI owners, and getting this book is surely cheaper and easier than buying up all of COMPUTE!'s back issues. I expect to have my nose in this book soon and often. Now let's see, what's this here about XIO?

Just Introduced
De Re ATARI (All About ATARI)
A Guide to 400/800 Home Computer
APX-90008

Order from:
Atari Program Exchange
P.O. Box 427
155 Moffett Park Drive
Sunnyvale, CA 94086
$19.95 + $2.50 shipping
CA residents add 6.5% tax
or phone 800-672-1430 in CA
800-538-1862 outside CA

KIDS KORNER

Do you think programming is dull and boring?

Well, try this, and you'll be surprised! Just type this program (exactly) into your ATARI and RUN it. Be sure there is a BASIC cartridge in the left front slot, though.

```
10 GRAPHICS 2
15 DIM NAM$(20)
20 ?"TYPE YOUR NAME"
25 INPUT NAM$
28 POSITION 10,0
30 SETCOLOR 0,0,0
32 POSITION 10,K
35 PRINT #6,NAM$
40 FOR I= 1 TO 15
45 FOR J = 1 TO 50
50 SETCOLOR 4,I, 8
55 NEXT J:NEXT I
58 K=K+1
60 GOTO 30
```

RUN the program. Type in your name and press RETURN.

Not bad, huh?

Everything that's fun on a computer has to be programmed by someone. Some programming is very complicated, but some is as simple as this. Of course, we know you don't understand what you did, but you can learn. Study your BASIC book, read computer books and magazines, share ideas with your friends, ask your teacher, pester a programmer, experiment.

ANTIC will help you discover the possibilities of your ATARI, and if you invent a good program, maybe we'll print it for other kids to use.

WORD PROCESSING COMPARISON

continued from page 5

TEXT WIZARD (TM)

BEST FEATURES:

The least expensive of the three word processors being discussed here, *Text Wizard* has to be labeled as the BEST BUY. It is an extremely powerful yet simple word processor, and has the added ability to write and edit BASIC programs. If you do any software development you will really appreciate the edit capability. For example, you can load a basic program, edit it, and print all or part of it to your printer.

Word processing commands are very logical and can be given at any time except during print. There is no menu, but every command ties to the keyboard in a fashion that is easy to remember. The console keys START, SELECT, AND OPTION, for example, are used a great deal. To print, save, or load a program you would press OPTION and either P, S, or L. Perhaps the nicest feature involves the INSERT mode. If you press the CONTROL key and the INSERT key at the same time, you enter this mode which allows you to insert letters, words, or even paragraphs. Text to the right of the cursor is "pushed" as you type which makes the job of adding text into the middle of a paragraph a simple matter.

WORST FEATURES:

Text Wizard makes no attempt to overcome the 40 column TV screen limitation imposed by Atari. This translates to "print it if you want to see how it will look" and wastes a lot of paper. Difficulty may also be experienced when using the header/footer commands on shorter than normal pages, but with some experimenting this can be compensated for. Although you would have to do a lot of typing to feel the pinch, *Text Wizard* uses more of your memory space than *Letter Perfect* by about 6000 bytes.

ATARI (TM) WORD PROCESSOR

BEST FEATURES:

Make no mistake, this is the most sophisticated and powerful of the three programs being compared. At the top of most lists would have to be the superior text formatting/viewing capabilities of the ATARI word processor. Of the three programs we are reviewing, this is the only one which does an excellent job of compensating for the 40 column screen limitation. By pressing the [SELECT] key, you are transferred to a special graphics mode which shows you what your document will look like when printed. Although you can't read it in this form (words have been replaced by lines) the image is very clear. It is from this mode that all of the form controls are indicated as well, and you can see the effect of your margin changes instantly. All other changes require that your text be re-formatted under program control. The reformatting, incidentally, is fascinating to watch the first few times, but on a long document can be time consuming. The trade off of reformatting versus printing to see what the results will be, softens the wait considerably.

The *Atari Word Processor* has a number of very sophisticated features that aren't found in many others in its price range ($150). It makes liberal use of the Atari scrolling and screen editing features which provides substantial freedom to view your text. It intentionally makes a backup copy of the last page that you "re-save" after editing. This allows you to reclaim it in the event you have a change of heart. It allows you to change capital letters to lower case, and vice-versa with a simple keystroke. You would have to see this one in action to belive its usefulness.

WORST FEATURES:

After using both *Letter Perfect* and *Text Wizard* for some time the *Atari Word Processor* seems very complicated even though great pains have been taken in the Atari tradition of human engineering. The documentation, which seems intimidating at first, is designed for the person with no computing experience and is demeaning in its simplicity for anyone other than the rank beginner. Time is the best teacher, though, and Atari has really done a commendable job of preparing the user to reap full benefits from their program. After the dust has settled, the one remaining shortcoming seems to be the numerous keystrokes needed to accomplish certain tasks. The *Atari Word Processor* is completely menu driven, with at least 5 levels of menu to contend with in certain cases. To format a document, for example takes five individual keystrokes if you want to include depagination. Printing can take four and so-on. Much of this results from the added features and functions that Atari has provided, so it is possible that there isn't a better/faster way. It just seems that there is so much "protect the user" philosophy built in to it that is sometimes gets in the way of efficient operation.

CONCLUSIONS

The best way to finish this off is to refer you to the chart of features that accompany this article, and to encourage you to try each of them at a local dealer. All three word processors are good, if not great. All have their strengths, and in fact I use all three depending on my purpose or goal. All three are good value for the money. If I had to choose one above the others, however, I would give the nod to *LJK*. They seem to be on the right track, and are tying their software together into a nice, albeit expensive, package. The database *(Data Perfect)* is now available as well as the ROM version of *Letter Perfect*. The latter, incidentally, sells for $250, which seems a bit high. It is an enhanced version of Version 2.0 and will be given a thorough review in the next issue, so be sure to check it out.

CUSTOMER SUPPORT

Atari is committed to its customers. They have tried to maintain the best support of any major computer manufacturer that I know of.

Unlike other manufacturers who have forsaken their users, Atari maintains toll free numbers for customer questions and problems.

In California
800-672-1430
Outside California
800-538-8547
In Hawaii and Alaska
800-536-1682

The APX (Atari Program Exchange) toll free line for ordering is:

In California
800-672-1430
Outside California
800-538-1862

!! COMING IN JUNE !!

PASCAL—It's finally here and we'll tell you about it.

PROFILES—Interview with Ihor Wolosenko, co-developer of File Manager 800 and founder of SYNAPSE Software.

PRODUCTS IN DEPTH—Investigation of communications hardware for your computer & telephone. A look at two major speech synthesis devices.

INSIDE ATARI—An interview with a surprise guest from Atari headquarters.

REGULAR FEATURES—Pilot, Memory Map (continued), Starting Line, Forth Factory, I/O Board (readers responses), Education Center, Kids Korner, and Tape Topics.

WATCH FOR—Comments on the West Coast Computer Faire. New software reviewed. More software in the public domain. Useful assembly language routines. GTIA—more graphics from this new chip. More on business and finance. Cassette tape users rejoice—help for you too.

!! WE ARE THE ATARI™ RESOURCE !!

Available to Subscribers Soon!!

PUBLIC DOMAIN SOFTWARE

Collected from all over the country

- ★ Great Value
- ★ Games
- ★ Utilities
- ★ Music
- ★ Education
- ★ Inexpensive Software
- ★ Software by Category

WATCH FOR THIS IN JUNE

User Groups Contact Us

Users Benefit from All the Clubs

ADVERTISERS

GET SERIOUS OR ELSE... JUST PLAY GAMES!

For Atari home computer

FILEMANAGER T.M.

The Atari 800 disk-based file management system.

FILEMANAGER T.M. is a powerful personal, as well as professional, tool. The full use of graphics and sound make FILEMANAGER T.M. an extremely easy-to-use file-keeping system.

FILEMANAGER T.M. provides:

* User defined formats
* Index on any field at any time
* Ultra fast sort
* Wild card search
* Up to 2000 characters per record
* Full screen input & edit
* Search on any combination of fields
* Format lists and mailing labels
* Single keystroke commands
* Full mathematical capability
* Computed, constant, and dollar fields
* Combine and convert existing files
* Elegant human engineering
* Database merge w/ Datasoft's Text Wizard
* Exclusive Datalock T.M. electronic key protection
* Compatable w/ Axlon's 128K RAMDISK
* And much, much more . . .

Use FILEMANAGER T.M. to create mailing lists, inventory and business records, customer lists, all customized to your own needs!

DISKMANAGER T.M.

Now you can maintain an inventory of all the programs in your growing library, on one disk. DISKMANAGER T.M. prints out labels (category, date and number) and alphabetized lists for each disk in your library.

The DISKMANAGER T.M. can automatically update your library when you add or delete programs. Fast machine language sorts allow you to locate any program in your library quickly and efficiently. DISKMANAGER T.M. offers full "wildcard" search functions as well as a variety of other professional features!

DODGE-RACER T.M.

Drive your car through a torturous maze avoiding the computer jam car out to get you. The better you get, the faster the action . . . And when you're ready, play two cars at once! Challenge the computer or your friends with 12 game options in this real-time game of skill and reflex.

PROTECTOR T.M. - by Mike Potter

You are the last hope! The alien Slime-hordes of Fraxullus are attacking your cities. One by one, the inhabitants are being snatched from the rooftops and dropped into an erupting volcano.

You must save these innocent victims from their gruesome destiny. Laser caves, Xytonic pulse-trackers, meteor storms and other dangers await you. It has fallen on your shoulders, you are the Protector.

There are six levels of lightning-fast machine language play in this hi-res arcade game.

CHICKEN T.M.

There's trouble in the barnyard. Ma Hen and a pesky old fox battle it out, as you try to save the local chicken population. Seems that the fox has found Ma Hen's eggs, and she is trying to save them from his deadly clutches. The action gets faster and faster as eggs turn into chicks, feathers fly, chickens squawk, and all bedlam breaks loose.

CHICKEN T.M. is a machine language, arcade game that will challenge and entertain even the most experienced arcade player.

Ask for details on all Synapse Software products at your local dealer.

synapse
S O F T W A R E

820 Coventry Road, Kensington, CA 94707
(415) 527–7751

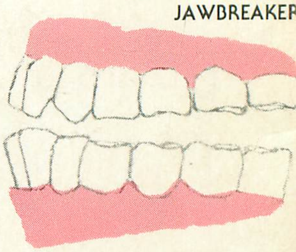


WE BRING YOU THE BEST IN ATARI GAMING


THE NEXT STEP


THRESHOLD

ON-LINE systems



JAWBREAKER

At ON-LINE SYSTEMS, we have worked hard to make the software that you want. Over the last 9 months, authors like John Harris and Bob Stewart have been making totally new software for you, and software specialists like Chris Iden and Warren Schwader have been bringing you the best software available from other computers. Their efforts have not gone unnoticed and more and more, ATARI owners are coming to us for their software needs.

— Here's why —

MOUSKATTACK

You are Larry Bain, ace plumber for RODENT ROOTER Plumbing service. Your job is to install pipe through the many levels of rat alley. One look at a super rat will tell you why it won't be easy. An original arcade game by JOHN HARRIS (author of Jawbreaker).

32K ATARI 400/800 Disk PRICE $34.95

CROSSFIRE

They're coming at you from all sides. If your going to make it, your going to have to move fast. This arcade game is totally original and highly addicting. By Jay Sullivan and Chris Iden.

32K ATARI 400/800 Disk PRICE $29.95

JAWBREAKER

Already an ATARI classic. John Harris has made a fantastic arcade game that is fun for the whole family. Can you eat your way through the candy store without getting caught by a bratty kid? Get JAWBREAKER and find out.

24K ATARI 400/800 Disk PRICE $29.95
16K ATARI 400/800 CASS PRICE $29.95


MISSION ASTEROID

HI-RES #0
MISSION ASTEROID

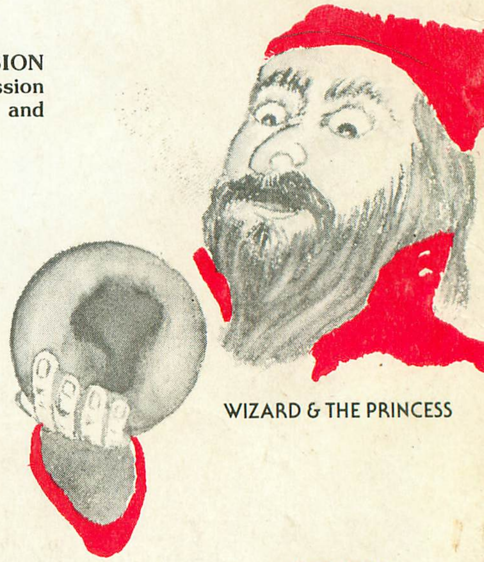
An introduction to our HI-RES adventure game series. MISSION ASTEROID brings you the sights of outer space and mission control in beautiful 21 color HI-RES graphics. By Ken and Roberta Williams and Chris Iden.

40K ATARI 400/800 Disk PRICE $24.95

HI-RES #2
WIZARD & THE PRINCESS

An adventure classic comes to the ATARI! The evil wizard Harlan has stolen away the princess Priscilla. Rescue her and half the kingdom of Serenia is yours. Beautiful 21 color HI-RES graphics. By Ken and Roberta Williams and Chris Iden.

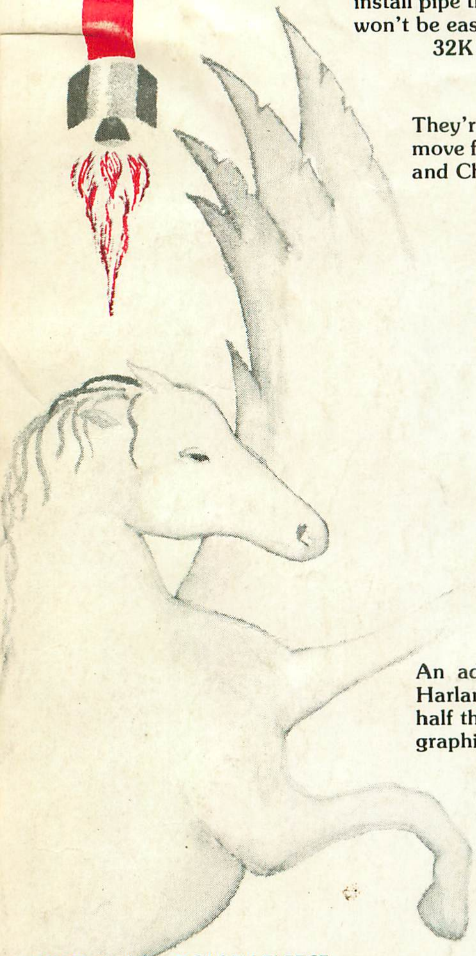
40K ATARI 400/800 Disk PRICE $32.95


WIZARD & THE PRINCESS

HI-RES #4
ULYSSES and the GOLDEN FLEECE

You are ULYSSES and the king has sent you to retrieve the golden fleece for his pleasure. To win the fleece, you will have to do battle with the powerful enemies from ancient mythology. Beautiful 21 color HI-RES Graphics. By Bob Davis, Ken Williams and Chris Iden.

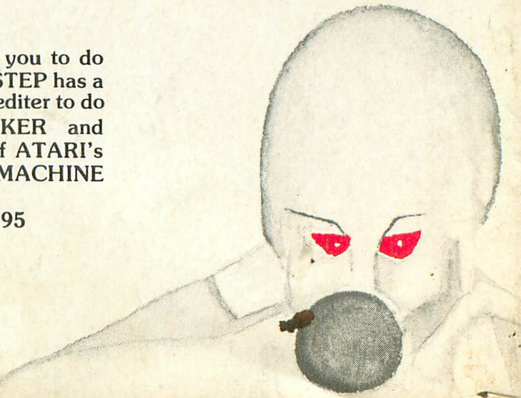
40K ATARI 400/800 Disk PRICE $34.95


ULYSSES and the GOLDEN FLEECE

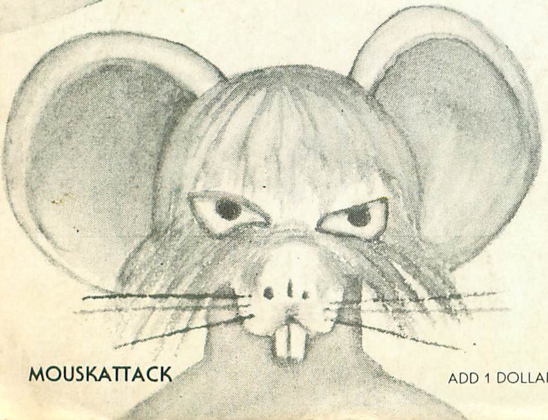
THE NEXT STEP

A powerful new graphics utility that will help you to do fantastic graphics on your ATARI. The NEXT STEP has a menu driven, joystick controlled, character set editer to do graphics like those found in JAWBREAKER and MOUSKATTACK, plus routines to mix any of ATARI's 14 graphics modes. A must for the BASIC or MACHINE LANGUAGE programmer.

32K ATARI 400/800 Disk PRICE $39.95


CROSSFIRE

ON-LINE SYSTEMS
36575 MUDGE RANCH RD
COARSEGOLD, CA 93614
(209) 683-6858


MOUSKATTACK

ADD 1 DOLLAR FOR SHIPPING

VISA, MASTERCARD, CHECK, COD